

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 4月18日

出 願 番 号

Application Number:

特願2001-120335

[ST.10/C]:

[JP2001-120335]

出 願 人

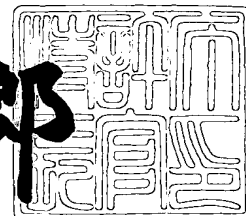
Applicant(s):

大森 聡

2003年 3月14日

特許庁長官
Commissioner,
Japan Patent Office

太田信一郎



出証番号 出証特2003-3017022

【書類名】 特許願

【整理番号】 2001A09

【提出日】 平成13年 4月18日

【あて先】 特許庁長官殿

【発明者】

 【住所又は居所】 埼玉県浦和市西堀4丁目11番7号627

 【氏名】 大森 聡

【特許出願人】

 【識別番号】 300000513

 【氏名又は名称】 大森 聡

【先の出願に基づく優先権主張】

 【出願番号】 特願2000-117343

 【出願日】 平成12年 4月19日

【先の出願に基づく優先権主張】

 【出願番号】 特願2000-149122

 【出願日】 平成12年 5月19日

【手数料の表示】

 【予納台帳番号】 095958

 【納付金額】 21,000円

【提出物件の目録】

 【物件名】 明細書 1

 【物件名】 図面 1

 【物件名】 要約書 1

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 ヌクレオチド等の配列情報の記録方法及び装置、前記配列情報の供給方法、前記配列情報を記録した記録媒体、並びに要約値の計算方法

【特許請求の範囲】

【請求項 1】 一列のヌクレオチドの配列情報の記録方法であって、
前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報を記録することを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 2】 請求項 1 記載の記録方法であって、
前記一列のヌクレオチドは 4 種類のヌクレオチドよりなり、
前記 4 種類のヌクレオチドを互いに異なる 6 ビット以下のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 3】 請求項 2 記載の記録方法であって、
前記 4 種類のヌクレオチドを互いに異なる 2 ビットのデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 4】 請求項 2、又は 3 記載の記録方法であって、
前記一列のヌクレオチドは、一つの DNA を構成する 1 対の重合体の鎖の内の 1 本の鎖の全部又は一部であり、
前記 4 種類のヌクレオチド中の互いに相補的な 2 対のヌクレオチドをそれぞれ互いにビット反転の関係にある 1 対のデータで表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 5】 請求項 2、又は 3 記載の記録方法であって、
前記一列のヌクレオチドは、一つの RNA を構成する一つの重合体の鎖の全部又は一部であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 6】 請求項 1 記載の記録方法であって、
前記一列のヌクレオチドの配列に関する情報を、前記配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 7】 請求項 6 記載の記録方法であって、

前記一列のヌクレオチドは 2 5 個以上のヌクレオチドの配列であり、

前記一列のヌクレオチドの配列に関する情報を 4 0 ビット以上で 1 9 2 ビット以下の長さの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 8】 請求項 7 記載の記録方法であって、

前記数学的な要約値は、前記一列のヌクレオチドの配列に対応するテキストデータ又は数値データに MD 5 ハッシュ関数、又は S H S ハッシュ関数の演算を施して得られることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 9】 請求項 1 ～ 6 の何れか一項記載の記録方法であって、

前記一列のヌクレオチドは遺伝子の少なくとも一部であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 1 0】 請求項 1 記載の記録方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し

、
複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報で前記一列のヌクレオチドの配列を表すことを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 1 1】 請求項 1 0 記載の記録方法であって、

複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交互に第 1 群の変換データ及び第 2 群の変換データに分けたとき、

前記第 1 の演算は、所定の整数 K を用いて前記第 1 群の変換データ、及び前記第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、

前記第 2 の演算は、複数列の前記変換データの各列の変換データに対する法 K のもとの和を求める演算であることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 1 2】 請求項 1 0、又は 1 1 記載の記録方法であって、

前記一列のヌクレオチドの配列を基準配列として、該基準配列の 2 組の前記シンドローム情報に対応させて、検査対象の一列のヌクレオチドの配列の 2 組のシンドローム情報を求め、

前記 4 組のシンドローム情報より前記基準配列に対する前記検査対象の一列のヌクレオチドの配列の相違部を求めることを特徴とするヌクレオチドの配列情報の記録方法。

【請求項 1 3】 一列のヌクレオチドの配列情報の記録装置であって、

一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置と、

該配列読み取り装置で読み取られた配列の情報をテキストデータとして第 1 ファイルに記録する第 1 記録手段と、

前記第 1 ファイルのテキストデータよりも少ないデータ量で、前記配列読み取り装置で読み取られた配列の情報を表し、該配列の情報を第 2 ファイルに記録する第 2 記録手段と

を有することを特徴とするヌクレオチドの配列情報の記録装置。

【請求項 1 4】 請求項 1 3 記載の記録装置であって、

前記第 2 記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオチドの配列を、該配列を表すテキストデータ又は数値データの数学的な要約値で表すことを特徴とするヌクレオチドの配列情報の記録装置。

【請求項 1 5】 請求項 1 3 記載の記録装置であって、

前記第 2 記録手段は、前記配列読み取り装置で読み取られた一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに

異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し

、
複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報を前記第 2 ファイルに記録すること
を特徴とするヌクレオチドの配列情報の記録装置。

【請求項 1 6】 一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、

前記一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、前記一列のヌクレオチドの配列に関する情報が記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 7】 請求項 1 6 記載の記録媒体であって、

前記一列のヌクレオチドは 2 5 個以上のヌクレオチドの配列であり、

前記一列のヌクレオチドの配列に関する情報は、4 0 ビット以上で 1 9 2 ビット以下の長さの数学的な要約値で前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 8】 請求項 1 6 記載の記録媒体であって、

前記一列のヌクレオチドの配列に対応するテキストデータを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し

、
複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求めておき、

前記一列のヌクレオチドの配列に関する情報は、前記第 1 組及び第 2 組のシンδροーム情報として前記記録媒体に記録されたことを特徴とするコンピュータ読み取り可能な記録媒体。

【請求項 1 9】 一列のヌクレオチドの配列情報の供給方法であって、

前記一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

前記一列のヌクレオチドの配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするヌクレオチドの配列情報の供給方法。

【請求項 2 0】 請求項 1 9 記載の供給方法であって、

前記一列のヌクレオチドは 2 5 個以上のヌクレオチドの配列であり、

前記数学的な要約値は、4 0 ビット以上で 1 9 2 ビット以下のデータであり、

前記供給者は、更に前記一列のヌクレオチドの所定の一部の配列の情報を通信回線を介して閲覧可能な状態にしておくことを特徴とするヌクレオチドの配列情報の供給方法。

【請求項 2 1】 請求項 1 9、又は 2 0 記載の供給方法であって、

前記供給者は、前記一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応する前記数値データを第 1 ファイルに記録して保持し、

前記供給者は、前記テキストデータ、又は前記数値データを、前記ヌクレオチドの配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報を第 2 ファイルに記録して保持し、

第 1 段階として前記ユーザは、前記供給者より前記第 2 ファイルに記録されている 2 組のシンδροーム情報を受け取り、

前記 2 組のシンδροーム情報に基づいて検査対象の一系列のヌクレオチドの配列の内の前記供給者の一系列のヌクレオチドの配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第 2 段階として前記ユーザは前記供給者より前記第 1 ファイルに記録されている前記テキストデータ、又は前記数値データの内の前記配列の復元ができない部分の情報の提供を要求することを特徴とするヌクレオチドの配列情報の供給方法。

【請求項 2 2】 一系列のアミノ酸の配列情報の記録方法であって、

前記一系列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、前記一系列のアミノ酸の配列に関する情報を記録することを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 3】 請求項 2 2 記載の記録方法であって、

前記一系列のアミノ酸は、一つのタンパク質を構成する 1 本のアミノ酸の鎖の全部又は一部であり、

前記一系列のアミノ酸の配列に対応するテキストデータを、20 種類のアミノ酸に対して互いに異なる 6 ビット以下のデータを割り当てることによって変換することを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 4】 請求項 2 2 記載の記録方法であって、

前記一系列のアミノ酸の配列に関する情報を、前記配列を表すテキストデータの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 5】 請求項 2 4 記載の記録方法であって、

前記一系列のアミノ酸は 25 個以上のアミノ酸の配列であり、

前記一系列のアミノ酸の配列に関する情報を 16 ビット以上で 192 ビット以下

の長さの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 6】 請求項 2 4、又は 2 5 記載の記録方法であって、

前記数学的な要約値は、前記一列のアミノ酸の配列に対応するテキストデータに MD 5 ハッシュ関数、又は S H S ハッシュ関数の演算を施して得られることを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 7】 請求項 2 2 記載の記録方法であって、

前記一列のアミノ酸の配列に対応するテキストデータを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、

前記部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに異なる 8 ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報で前記一列のアミノ酸の配列を表すことを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 8】 請求項 2 7 記載の記録方法であって、

複数行の前記変換データの各行の変換データをそれぞれ前記非配列方向に交互に第 1 群の変換データ及び第 2 群の変換データに分けたとき、

前記第 1 の演算は、所定の整数 K を用いて前記第 1 群の変換データ、及び前記第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、

前記第 2 の演算は、複数列の前記変換データの各列の変換データに対する法 K のもとの和を求める演算であることを特徴とするアミノ酸の配列情報の記録方法。

【請求項 2 9】 一列のアミノ酸の配列情報の記録装置であって、

一つのタンパク質の少なくとも一部に含まれる一列のアミノ酸の配列情報をテキストデータとして第 1 ファイルに記録する第 1 記録手段と、

前記第 1 ファイルのテキストデータよりも少ないデータ量で、前記一列のアミノ酸の配列の情報を表し、該配列の情報を第 2 ファイルに記録する第 2 記録手段と

を有することを特徴とするアミノ酸の配列情報の記録装置。

【請求項 3 0】 請求項 2 9 記載の記録装置であって、

前記第 2 記録手段は、前記一列のアミノ酸の配列を、該配列を表すテキストデータの数学的な要約値で表すことを特徴とするアミノ酸の配列情報の記録装置。

【請求項 3 1】 一列のアミノ酸の配列情報の供給方法であって、

前記一列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる 8 ビット以下の数値データを割り当てることによって前記テキストデータを変換して得られる数値データを保持する供給者が、

前記一列のアミノ酸の配列の長さの情報、及び前記配列を表すテキストデータ又は前記数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、

前記通信回線を介して前記配列の長さの情報及び前記数学的な要約値の情報を閲覧したユーザより、前記テキストデータ又は前記数値データの少なくとも一部の情報に対する取得要求が前記供給者に届いた後に、

前記供給者が前記ユーザに前記テキストデータ又は前記数値データの少なくとも一部の情報を供給することを特徴とするアミノ酸の配列情報の供給方法。

【請求項 3 2】 請求項 3 1 記載の供給方法であって、

前記一列のアミノ酸は 2 5 個以上のアミノ酸の配列であり、

前記数学的な要約値は、1 6 ビット以上で 1 9 2 ビット以下のデータであることを特徴とするアミノ酸の配列情報の供給方法。

【請求項 3 3】 請求項 3 1、又は 3 2 記載の供給方法であって、

前記供給者は、前記一列のアミノ酸の配列に対応するテキストデータ、又はこれに対応する前記数値データを第 1 ファイルに記録して保持し、

前記供給者は、前記テキストデータ、又は前記数値データを、前記アミノ酸の配列方向に複数行で、かつ前記配列方向に交差する非配列方向に複数列の部分データに分割し、

前記部分データを、それぞれ複数種類のアミノ酸に対して互いに異なる 8 ビット以下の数値データを割り当てることによって変換データに変換し、

複数行の前記変換データに各行毎に前記非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、

複数列の前記変換データに各列毎に前記配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求め、

前記第 1 組及び第 2 組のシンδροーム情報を第 2 ファイルに記録して保持し、

第 1 段階として前記ユーザは、前記供給者より前記第 2 ファイルに記録されている 2 組のシンδροーム情報を受け取り、

前記 2 組のシンδροーム情報に基づいて検査対象の系列のアミノ酸の配列の内の前記供給者の系列のアミノ酸の配列との相違部を特定し、

該相違部の配列の復元ができない場合に、第 2 段階として前記ユーザは前記供給者より前記第 1 ファイルに記録されている前記テキストデータ、又は前記数値データの情報の提供を前記供給者に要求することを特徴とするアミノ酸の配列情報の供給方法。

【請求項 3 4】 一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、

前記一つ又は複数のファイルに記録されたデータの中で所定のコードを無視して要約値を計算することを特徴とする要約値の計算方法。

【請求項 3 5】 請求項 3 4 記載の要約値の計算方法であって、

前記無視する所定のコードは、数字コード、スペースコード、及び改行コードであることを特徴とする要約値の計算方法。

【請求項 3 6】 請求項 3 4 記載の要約値の計算方法であって、

前記無視する所定のコードは、同一又は互いに異なる 2 組のコード、及びこれら 2 組のコードに挟まれたデータであることを特徴とする要約値の計算方法。

【請求項 3 7】 請求項 3 4 記載の要約値の計算方法であって、

前記一つ又は複数のファイルから 1 文字分のコードデータを読み出す毎に、

該読み出されたコードデータが前記所定のコードであるときには、該読み出されたコードデータを無視して、次の 1 文字分のコードデータの読み出しを行い、

該読み出しによって得られた前記所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うことを特徴とする要約値の計算方法。

【請求項 3 8】 一連のテキストデータの要約値を計算するための要約値の計算方法であって、

前記一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、前記所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、

前記複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、

該複数のファイルに記録されたテキストデータから分割の順序に従って順次要約値を計算することを特徴とする要約値の計算方法。

【請求項 3 9】 請求項 3 8 記載の要約値の計算方法であって、

前記所定個数ずつのコードデータ、及び前記所定個数よりも少ない個数のコードデータからは、所定のコードデータが除外されていることを特徴とする要約値の計算方法。

【請求項 4 0】 請求項 3 9 記載の要約値の計算方法であって、

前記所定個数は、要約値を計算する際のデータ量の単位に応じて定められることを特徴とする要約値の計算方法。

【発明の詳細な説明】

【 0 0 0 1 】

【発明の属する技術分野】

本発明は、例えば DNA（デオキシリボ核酸：deoxyribonucleic acid）又は RNA（リボ核酸：ribonucleic acid）等の核酸の少なくとも一部を構成する一列のヌクレオチドの配列情報、又はタンパク質の少なくとも一部を構成する一列のアミノ酸の配列情報の記録方法及び装置に関する。更に本発明は、その配列情報を供給するためのビジネスモデルとして好適な配列情報の供給方法、その配列情報を記録したコンピュータ読み取り可能な記録媒体、及びその記録方法を実施する際に使用できる要約値の計算方法に関する。

【 0 0 0 2 】

【従来の技術】

人間、及び他の生物（動物、植物、微生物等）のDNAを構成する1対のヌクレオチドの鎖（又は塩基の鎖）の配列情報の解読が世界的に行われている。この場合、従来よりDNAを構成する4種類のヌクレオチドは、塩基としてアデニンを含むヌクレオチド、グアニンを含むヌクレオチド、シトシンを含むヌクレオチド、及びチミンを含むヌクレオチドにそれぞれ文字A、G、C、及びTを割り当てることによって、それぞれ1バイト（＝8ビット）のテキストデータで表わされている。その結果として一つのDNAの配列は、それを構成する1対の重合体の鎖の内の一方向の鎖のヌクレオチド（n個とする）の配列を順次文字A、G、C、T（又はa、g、c、t）の何れかで表すことによって、nバイトのテキストデータで表されていた。同様に、一つのRNAを構成する1本のn個のヌクレオチドの配列は、チミンを含むヌクレオチドの代わりにウラシルを含むヌクレオチドに文字U（又はu）を割り当てることによって、nバイトのテキストデータで表されていた。

【 0 0 0 3 】

これに関して、例えば人間の最も大きい第1染色体中のDNAの配列は、約2億5千万個のヌクレオチドの配列であり、最も小さい第22染色体中のDNAの配列は、約5000万個のヌクレオチドの配列であるため、人間の各染色体中のDNAの配列は、約250Mバイト～50Mバイトのテキストデータで表すことができる。更に、一人の人間の全部のDNA情報（ゲノム）は、約30億個のヌクレオチドの配列で表すことができるため、そのゲノムは、約3Gバイトのテキストデータで記録することができる。なお、それらのテキストデータに対して通常のファイル圧縮技術を適用することによって、それらのテキストデータは、例えば元のデータの50％程度の圧縮ファイルとしても記録、又は送信することができる。

【 0 0 0 4 】

また、DNAの配列の解読に続いて、DNA中の多数の遺伝子の情報に基づいてそれぞれ合成されるタンパク質の機能の研究も広く行われている。この場合、

タンパク質を構成する 20 種類のアミノ酸は、三文字表記 (3-Letter Code) ではそれぞれ 3 文字 (例えば A l a, C y s, G l u 等) のテキストデータで表され、一文字表記 (1-Letter Code) ではそれぞれ 1 文字のテキストデータ (例えば A, C, E 等) で表されるため、n 個のアミノ酸よりなるタンパク質の配列は、n バイトのテキストデータで表すことができる。そして、種々のタンパク質は、それらのアミノ酸が約 20 個～約 1000 個程度所定の順序で配列されたものであるため、それらのタンパク質の配列は、最大でも約 1 k バイト程度のテキストデータで記録することができる。また、例えば人間の遺伝子の総数は約 3 万個と言われており、タンパク質は理論的なものも含めて約 10 万種類の存在が可能であると言われている。

【 0 0 0 5 】

【発明が解決しようとする課題】

上記の如く例えば一人の人間の DNA 情報をテキストデータで記録するためには、全部で 3 G バイト程度の記憶容量が必要であり、仮に通常の圧縮ファイルの技術を適用しても 1 G バイト程度の記憶容量が必要である。また、人間以外の大腸菌や各種ウイルス等の DNA 情報も解析されて次第に公開されるようになっていくが、これらの DNA 情報をテキストデータの形で多く集めると、数 100 M バイト程度の記憶容量が必要である。これは RNA の配列情報についても同様である。

【 0 0 0 6 】

このように人間又は他の生物の DNA 情報をテキストデータ、又はこの通常の圧縮ファイルの形で記録するものとする、例えば 1 枚の記憶容量が 5 G バイト程度の DVD-ROM (digital video disc-ROM) ディスクのように膨大な記憶容量を持つ記録媒体が必要である。更に、その DNA 情報を利用する場合にその記録媒体からの読み出し時間が長くなり、処理時間が長くなるという不都合がある。

【 0 0 0 7 】

また、現状の一般の通信回線の通信速度は、最大で 1 M b p s 程度であるため、例えば 1 G バイト程度の DNA 情報をその通信回線を介して送信するものとす

ると、送信時間は最短でも約2時間程度となり、あまり実用的ではない。特に最近はそのDNA情報をデジタルの携帯電話システムを介して送信する場合も考えられるが、現在の携帯電話システムの通信速度はせいぜい100kbp/s程度であるため、少なくとも人間のDNA情報の伝送で使用することは困難である。

【0008】

次に、例えば或る微生物のDNA中の遺伝子について複数の研究者が並行して研究するような場合に、複数の研究者が保有している標準となるDNAのヌクレオチドの配列の同一性をどのように保証するのかという問題がある。即ち、そのDNAのヌクレオチドの配列が例えば数Mバイト（文字数で数100万文字）程度のテキストデータで記録されている場合に、複数の研究者が互いに自分のテキストデータと他人のテキストデータとの同一性（完全一致性）を短時間に確認するのは必ずしも容易ではない。

【0009】

これに関連して、例えば人間又は他の生物のDNA情報の利用方法としては、標準的なDNAの配列と、検査対象のDNAの配列との間の相違する部分をサーチする場合は考えられる。これは、いわゆるSNP（一塩基変位多型：Single Nucleotide Polymorphism）の可能性を検査するような場合に必要になると考えられる。しかしながら、両方のDNAのヌクレオチドの配列がそれぞれ膨大なテキストデータで表わされている場合に、それら2つのテキストデータを比較して相違点を検出するにはかなりの長い時間が必要となり、検査時間が長くなるという不都合がある。

【0010】

更に、人間又は他の生物のDNA情報を製薬会社の研究者等のユーザに提供するビジネスも行われつつあるが、この場合に、複数の情報供給者間で重複した情報の提供をできるだけ避けることが望ましい。このためには、複数の情報供給者間で、DNAのヌクレオチドの全体の配列情報を公開することなく、ヌクレオチドの配列の同一性を容易に確認できるようにすることが望ましい。更に、情報供給者が例えば通信回線を介してDNA情報をユーザに提供する場合には、できるだけ少ない情報量で、即ち短い送信時間で必要な情報をユーザに提供できるビジ

ネスモデルが必要である。また、ユーザ側では、提供されたDNA情報に伝送エラー等が無いかどうかを容易に確認できることが望ましい。上記の各課題はRNAのヌクレオチドの配列情報についても同様に当てはまるものである。

【 0 0 1 1 】

更に、一つのタンパク質のアミノ酸の配列は、最大でも約1kバイト程度のテキストデータで記録することができるが、タンパク質の種類は理論的に約10万個程度にもなるため、全部のタンパク質の配列情報をテキストデータで表すと、全部のDNAの配列情報程度の膨大な量となる。従って、個々のタンパク質の配列は、できるだけ少ない情報量で記録できることが望ましい。また、2つのタンパク質の配列情報の同一性を容易に確認できるシステムも必要である。

【 0 0 1 2 】

本発明は斯かる点に鑑み、核酸中の一列のヌクレオチドの配列情報、又はタンパク質中の一列のアミノ酸の配列情報をできるだけ少ないデータ量で記録できる記録方法及び記録装置を提供することを第1の目的とする。

また、本発明は、2つのヌクレオチドの配列情報同士、又は2つのアミノ酸の配列情報同士の同一性を少ないデータ量で高精度に確認できる記録方法及び記録装置を提供することを第2の目的とする。

【 0 0 1 3 】

更に本発明は、2つのヌクレオチドの配列情報の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元できる記録方法及び記録装置を提供することを第3の目的とする。

また、本発明は、一列のヌクレオチドの配列情報、又は一列のアミノ酸の配列情報を少ないデータ量でユーザに提供できるビジネスモデルを提供することを第4の目的とする。

【 0 0 1 4 】

更に本発明は、そのビジネスモデルにおいて、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性、又は相違する部分を少ないデータ量で容易に確認できるようにすることをも目的とする。

また、本発明は、ヌクレオチドの配列情報が少ないデータ量で記録されたコン

ピュータ読み取り可能な記録媒体を提供することをも目的とする。

【 0 0 1 5 】

また、本発明は、ヌクレオチド又はアミノ酸の配列情報を記録する場合に使用できる効率的な要約値の計算方法を提供することを目的とする。

【 0 0 1 6 】

【課題を解決するための手段】

本発明によるヌクレオチドの配列情報の記録方法は、一列のヌクレオチドの配列情報の記録方法であって、その一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一列のヌクレオチドの配列に関する情報を記録するものである。

【 0 0 1 7 】

斯かる本発明によれば、その一列のヌクレオチドは、例えばDNA (deoxyribonucleic acid) を構成する1対の重合体の鎖の一方の鎖の少なくとも一部、又はRNA (ribonucleic acid) を構成する1列の重合体の鎖の少なくとも一部である。そして、その一列のヌクレオチドの配列は、各ヌクレオチドに含まれる塩基の配列ともみなすことができる。そのヌクレオチドの配列が、そのテキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、DVD-ROMのような大容量の媒体の他に、CD-ROM、及びフラッシュROMのような小容量でも通常のコンピュータで手軽に再生できる媒体を使用できる。

【 0 0 1 8 】

更に、少ないデータ量の配列情報であれば、通信回線を介して短時間に送信できるため、実質的に安価に配列情報の供給を行うことが可能となる。

本発明において、その一列のヌクレオチドは4種類のヌクレオチドよりなり、その4種類のヌクレオチドを互いに異なる6ビット以下のデータで表すことが望ましい。テキストデータ形式では、各ヌクレオチドは、それぞれ8ビットのアスキーコード (ASCII CODE)、即ち文字A, G, C, T (又はU) の何れかで表されるため、各ヌクレオチドを6ビット以下のデータで表すことによって、データ量を減らすことができる。

【 0 0 1 9 】

なお、テキストデータが記録されたファイルが通常の圧縮技術（Z I P ファイル、L H A ファイル等）で圧縮できるように、本発明のデータが記録されたファイルも更に通常の圧縮技術で圧縮して記録できることは言うまでもない。但し、圧縮されたファイルを使用する場合には、解凍作業が必要になり、最終的には元のファイルを復元する必要があるため、元のファイル自体のデータ量を減らしておくことは極めて有効である。

【 0 0 2 0 】

また、その4種類のヌクレオチドを互いに異なる2ビットのデータで表すことが望ましい。2ビットのデータによって、最も少ないデータ量で4種類のヌクレオチド（又は塩基）を表すことができる。

また、その一列のヌクレオチドが、一つのDNAを構成する1対の重合体の鎖の内の1本の鎖の全部又は一部であるときに、その4種類のヌクレオチド中の互いに相補的な2対のヌクレオチドをそれぞれ互いにビット反転の関係にある1対のデータで表すことが望ましい。互いに相補的な2対のヌクレオチドとは、互いに相補的な2対の塩基と実質的に同じ意味である。ここで、2進数で表現した数 k を $\text{bin}(k)$ として、例えば図2のDNA（5）に示すように、アデニンを含むヌクレオチド（7A）を $\text{bin}(00)$ で表したとき、それに対して相補的なチミンを含むヌクレオチド（7T）を $\text{bin}(11)$ で表す。更に、グアニンを含むヌクレオチド（7G）を $\text{bin}(01)$ で表したとき、それに対して相補的なシトシンを含むヌクレオチド（7C）を $\text{bin}(10)$ で表す。この結果、DNA（5）の一方のヌクレオチドの鎖（6A）が $\text{bin}(0001101111\cdots)$ （=BNAとする）で表されて、それと相補的な他方のヌクレオチドの鎖（6B）に対応する2進数のデータBNBは、コンピュータによって2進数BNAをビット毎に反転するだけで極めて高速に求めることができる。

【 0 0 2 1 】

次に、本発明において、より具体的な第1の記録方法は、その一列のヌクレオチドの配列に関する情報を、その配列を表すテキストデータ又は数値データの数学的な要約値（message digest）で表すものである。この数学的な要約値は、暗

号理論において、送信ファイルの作成者の本人確認を行うために、送信ファイルに所定のハッシュ関数を施すことによって得られる値と数学的には同等のものである。しかしながら、本発明においては、一列のヌクレオチドの配列を表すデータ（原データ）の要約値を、例えば最先の解読者の主張や、2つの膨大な原データの同一性の確認に使用する点が本質的に異なっている。即ち、或るDNAのヌクレオチドの配列を最初に解読した者が、その配列を示す原データの要約値を例えばインターネット上で公開することによって、原データを公開することなく最先に解読したことを主張できる。また、例えば情報供給者からDNAの配列情報を購入したユーザは、購入した配列情報の要約値を、例えばインターネット上で公開されているそのDNAの要約値と比較することによって、購入した配列情報の同一性を高い確率で確認できる。更に、複数の研究者が同一のDNAについて研究を行う場合に、各研究者が保持しているDNAのヌクレオチドの膨大な配列情報の長さ、及び要約値を求め、これらを比較することによって、研究対象の同一性を容易に高い確率で確認することができる。

【 0 0 2 2 】

この場合、その一列のヌクレオチドが25個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報を40ビット以上で192ビット以下の長さの数学的な要約値で表すことが望ましい。25個以上のヌクレオチドの配列のテキストデータは、200ビット（ $= 25 \cdot 8$ ビット）以上になるため、その要約値を192ビット以下とすることで、テキストデータよりも少ないデータ量となる。また、特に処理単位が64ビットのコンピュータを使用する場合には、要約値の長さは64ビットの倍数、即ち64ビット、128ビット、又は192ビットが望ましいと考えられる。

【 0 0 2 3 】

また、例えば将来的に全人類のDNAの配列情報を必要に応じて解読するような状況を想定して、世界人口を100億人程度と仮定すると、そのDNAの配列情報は約 10^{10} 通りにもなる。更に、安全係数を100倍程度とすると、その要約値は、 10^{12} （ $= 10^{10} \cdot 100$ ）通り、即ち約 $2^{39.86}$ 通り以上の値を取る必要がある。このためには、その要約値を40ビット以上の長さとするればよい。

これによって、2つのDNA又はRNAの配列情報同士の同一性を 10^{-12} 以上の精度で高精度に確認できる。

【0024】

更に、その数学的な要約値は、その一列のヌクレオチドの配列に対応するテキストデータ又は数値データにMD5ハッシュ関数、又はSHS (Secure Hash Standard) ハッシュ関数の演算を施して得ることができる。この場合、MD5ハッシュ関数は、高速演算が可能であると共に、得られる要約値が128ビットであり、通常のコンピュータで処理し易い利点がある。一方、SHSハッシュ関数は、元のデータの推定がより困難であるが、得られる要約値が160ビットと、通常のDNA又はRNAのヌクレオチドの配列の表現に関しては必要以上に長いと考えられる。従って、通常のヌクレオチドの配列の表現については、MD5ハッシュ関数がより実用的と考えられる。

【0025】

また、暗号理論で使用されるハッシュ関数は、送信ファイルの内容が推定されないように、かつ内容の衝突の確率が極めて低くなるように設計されるため、その要約値は例えば最低でも128ビット程度の長さが必要とされると共に、複雑な演算が繰り返して実行される。これに対して本発明で使用するハッシュ関数は、通常の互いに異なるヌクレオチドの配列に対してほぼ衝突が無ければよいため、あまり複雑な演算を繰り返して行う必要は無いと考えられる。但し、通常の暗号理論で要約値の演算対象となるファイルは、せいぜい1Mバイト程度の長さであるのに対して、本発明で使用するハッシュ関数の演算対象は、例えば人間のDNAのヌクレオチドの配列とすると、100Mバイト程度にも達する膨大なデータのファイルである。そこで、本発明で使用するハッシュ関数（ハッシュ演算プログラム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

【0026】

次に、本発明において、より具体的な第2の記録方法は、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータT（

i, j) に分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データ $A(i, j)$ に変換し、複数行のその変換データに各行毎にその非配列方向に第 1 の演算を施して第 1 組のシンδροーム (syndrome) 情報 $B_1(i)$, $B_2(i)$ を求めると共に、複数列のその変換データに各列毎にその配列方向に第 2 の演算を施して第 2 組のシンδροーム情報 $C(j)$ を求め、その第 1 組及び第 2 組のシンδροーム情報でその一列のヌクレオチドの配列を表すものである。

【 0 0 2 7 】

本発明においては、テキストデータを複数行で複数列の部分テキストデータに分割した後に、各部分テキストデータをそれぞれ変換データに変換しているが、これは予めそのテキストデータを一列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えば図 7 に示す部分テキストデータ $T(i, j)$ を集めたテキストデータの情報の大部分を、例えば図 9 に示す第 1 組のシンδροーム情報 $B_1(i)$, $B_2(i)$ 、及び第 2 組のシンδροーム情報 $C(j)$ で表すことができる。具体的に、図 7 のテキストデータを配列方向に N 個 ($i = 1 \sim N$) で、非配列方向に M 個 ($j = 1 \sim M$) の部分テキストデータ $T(i, j)$ に分割し、各部分テキストデータ $T(i, j)$ が 16 個分のヌクレオチドのテキストデータを含むものとする、元のテキストデータのデータ量 DT_1 は、以下のようになる。

【 0 0 2 8 】

$$DT_1 = 16 \cdot N \cdot M \text{ (バイト)} \quad \dots (1)$$

更に、各ヌクレオチドを 2 ビットのデータで表すものとする、各部分テキストデータ $T(i, j)$ は、それぞれ 32 ビットの変換データ $A(i, j)$ に変換され、シンδροーム情報 $B_1(i)$, $B_2(i)$, $C(j)$ も 32 ビットのデータとなる。また、非配列方向のシンδροーム情報が 2 列 $B_1(i)$, $B_2(i)$ あるとすると、シンδροーム情報のデータ量 DS_1 は、以下のようになる。

【 0 0 2 9 】

$$\begin{aligned} DS_1 &= 32 (2 \cdot N + M) \text{ (ビット)} \\ &= 4 (2 \cdot N + M) \text{ (バイト)} \quad \dots (2) \end{aligned}$$

従って、仮に $N = 64$, $M = 128$ とすると、(1) 式及び (2) 式よりデータ量 $DT1$, $DS1$ は以下ようになる。

$$DT1 = 131072 \text{ (バイト)} \doteq 130 \text{ k バイト} \quad \dots (3)$$

$$DS1 = 1024 \text{ (バイト)} = DT1 / 128 \quad \dots (4)$$

従って、シンドローム情報のデータ量は、元のテキストデータのデータ量のほぼ $1/100$ 程度に圧縮できる。この場合、例えば人間の 1 本の染色体の DNA の配列は、50Mバイト～250Mバイト程度のテキストデータで表されるため、予めそのテキストデータを 500 個～2500 個程度のブロックに分割し、各ブロック毎にシンドローム情報を求めることによって、全部のシンドローム情報のデータ量はそのテキストデータのほぼ $1/100$ 程度、即ち 500 k バイト～2.5Mバイト程度に圧縮される。この程度のデータ量であれば、例えば携帯電話システムのような低速の通信回線を介しても短時間に送信できると共に、DVD-ROM よりも容量の少ない CD-ROM 等の記録媒体にも余裕を持って記録することができる。

【0030】

この場合、複数行のその変換データの各行の変換データをそれぞれその非配列方向に交互に第 1 群の変換データ (例えば奇数番目の変換データ $A(i, 1)$, $A(i, 3)$, ...) 及び第 2 群の変換データ (例えば偶数番目の変換データ $A(i, 2)$, $A(i, 4)$, ...) に分けたとき、その第 1 の演算は、所定の整数 K を用いてその第 1 群の変換データ、及びその第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、その第 2 の演算は、複数列のその変換データの各列の変換データに対する法 K のもとの和を求める演算である。その変換データ $A(i, j)$ を s ビット (例えば $s = 32$, $s = 64$ 等) とすると、その整数 K は一例として次のようになる。

【0031】

$$K = 2^s \quad \dots (5)$$

通常のコンピュータでは、その法 K のもとの和演算は極めて高速に実行することができる。

また、その一列のヌクレオチドの配列を基準配列として、この基準配列の 2 組

のそのシンドローム情報 ($B1(i)$, $B2(i)$, $C(j)$) に対応させて、検査対象の一行のヌクレオチドの配列 ($TF(i, j)$) の 2 組のシンドローム情報 ($B1F(i)$, $B2F(i)$, $CF(j)$) を求め、その 4 組のシンドローム情報よりその基準配列に対するその検査対象の一行のヌクレオチドの配列の相違部を求めることが望ましい。例えば図 7 の配列を基準配列、図 10 の配列を検査対象の配列として、図 7 の基準配列のシンドローム情報が図 8 に、図 10 の配列のシンドローム情報が図 11 に表されている。このとき、図 8 のシンドローム情報 ($B1(i)$, $B2(i)$, $C(j)$) に対して、図 11 のシンドローム情報 ($B1F(i)$, $B2F(i)$, $CF(j)$) は、 $B1F(1)$, $B2F(4)$, $CF(16)$, $CF(17)$ の値が異なるため、それらの交点として、図 10 の部分テキストデータ $TF(4, 16)$, $TF(1, 17)$ が図 7 の基準配列と異なっていることを検出できる。即ち、4 組のシンドローム情報を比較することによって、少ないデータ量の比較で、検査対象の配列のどの部分テキストデータが基準配列と異なっているかを検出できる。

【 0 0 3 2 】

この際に、基準配列と異なっている部分をエラーコード (error code) と呼ぶと、エラーコードが部分テキストデータの各行、又は各列に一つである場合には、それら 4 組のシンドローム情報、及びその検査対象のエラーコードに対応する変換データの法 K の加減算より、基準配列の変換データ $A(4, 16)$, $A(1, 17)$ 、ひいては部分テキストデータ $T(4, 16)$, $T(1, 17)$ が正確に復元できる。従って、例えば遺伝子中の一つの塩基 (ヌクレオチド) だけが異なる SNP (一塩基変位多型: Single Nucleotide Polymorphism) は本発明によって容易に検出できると共に、それに対応する正常な配列も容易に復元できる。

【 0 0 3 3 】

なお、図 10 の場合のように隣接する 2 つの列の部分テキストデータ $TF(4, 16)$, $TF(1, 17)$ に跨るような長いエラーコード (以下、「バーストエラー (burst error)」と呼ぶ) が存在する場合に、非配列方向のシンドロームが各行に 1 つ (即ち、 $B1F(i)$ と $B2F(i)$ との和) のみであるとする、1 行中の 2 箇所の部分テキストデータ、及び 1 列中の 2 箇所の部分テキストデ

ータにエラーコードが検出されてしまう。従って、エラーコードの位置の誤検出が生じて、それに対応する基準配列の復元も困難となる。これに対して本発明のように各行で2つのシンδροーム情報を求めることによって、バーストエラーの検出及び復元を正確に行うことができる。なお、各行で2つのシンδροーム情報を求める代わりに、配列方向（各列）で例えば前半分と後半分との2群の変換データに対して2つのシンδροーム情報を求めるようにしてもよく、どちらを採用するかは全体のデータ量が少なくなるように選択すればよい。

【 0 0 3 4 】

次に、本発明の記録装置は、一列のヌクレオチドの配列情報の記録装置であって、一つの核酸の少なくとも一部に含まれる一列のヌクレオチドの配列情報を読み取る配列読み取り装置（4）と、この配列読み取り装置で読み取られた配列の情報をテキストデータとして第1ファイル（19）に記録する第1記録手段（ステップ102～104）と、その第1ファイルのテキストデータよりも少ないデータ量で、その配列読み取り装置で読み取られた配列の情報を表し、この配列の情報を第2ファイル（20，21）に記録する第2記録手段（ステップ105～107）とを有するものである。これによって、本発明の配列情報の記録方法が実施できる。

【 0 0 3 5 】

この場合、その第2記録手段は、一例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列を、この配列を表すテキストデータ又は数値データの数学的な要約値で表すものである。

また、その第2記録手段は、別の例としてその配列読み取り装置で読み取られた一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報

を求め、その第 1 組及び第 2 組のシンδροーム情報をその第 2 ファイルに記録するものである。

【 0 0 3 6 】

また、本発明の記録媒体は、一列のヌクレオチドの配列情報を記録したコンピュータ読み取り可能な記録媒体であって、その一列のヌクレオチドの配列に対応するテキストデータよりも少ないデータ量で、その一列のヌクレオチドの配列に関する情報が記録されたものである。本発明によれば、例えば人間の DNA のヌクレオチドの配列情報を、少ないデータ量で記録できるため、記録媒体として CD-ROM, CD-R 等の使い勝手の良い媒体を使用できる。

【 0 0 3 7 】

この場合、その一列のヌクレオチドが 2 5 個以上のヌクレオチドの配列であるときに、その一列のヌクレオチドの配列に関する情報は、一例として 4 0 ビット以上で 1 9 2 ビット以下の長さの数学的な要約値でその記録媒体に記録されるものである。この場合には、記録媒体としてフレキシブルディスクであっても使用できる。

【 0 0 3 8 】

また、別の例として、その一列のヌクレオチドの配列に対応するテキストデータを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のヌクレオチドに対して互いに異なる 6 ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第 1 の演算を施して第 1 組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第 2 の演算を施して第 2 組のシンδροーム情報を求めておき、その一列のヌクレオチドの配列に関する情報は、その第 1 組及び第 2 組のシンδροーム情報としてその記録媒体に記録される。この記録媒体を用いることによって、例えば 2 つの試料のヌクレオチドの配列の相違する部分の位置の検出ができると共に、その相違する部分が少ない場合にはそれに対応する配列の復元を行うことができる。

【 0 0 3 9 】

次に、本発明の配列情報の供給方法は、一列のヌクレオチドの配列情報の供給方法であって、その一列のヌクレオチドの配列に対応するテキストデータ、又は複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者（2A）が、その一列のヌクレオチドの配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線（1）を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧したユーザ（2B）より、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

【0040】

この供給方法は、上記の本発明のヌクレオチドの配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る生物XのDNAのヌクレオチドの配列を最初に解読した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にその生物XのDNAの配列を解読したことを主張できる。更に、ユーザが同じ配列情報を異なる供給者から誤って購入することも防止できる。

【0041】

また、或るユーザが、その供給者よりその生物XのDNAの配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

【0042】

この場合、その一列のヌクレオチドは25個以上のヌクレオチドの配列である

ときに、一例として、その数学的な要約値は、40ビット以上で192ビット以下のデータであり、その供給者は、更にその一列のヌクレオチドの所定の一部の配列の情報をその通信回線を介して閲覧可能な状態にしておくことが望ましい。その要約値、及びその配列の長さの他に、そのように例えばその配列の先頭の8個程度、及び後端の8個程度の配列を比較することによって、同一性の確認をより高精度に行うことができる。

【 0 0 4 3 】

また、その供給者は、その一列のヌクレオチドの配列に対応するテキストデータ、又はこれに対応するその数値データを第1ファイル（19）に記録して保持し、その供給者は、そのテキストデータ、又はその数値データを、そのヌクレオチドの配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分データに分割し、その部分データを、それぞれ複数種類のヌクレオチドに対して互いに異なる6ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報を求め、その第1組及び第2組のシンδροーム情報を第2ファイル（20）に記録して保持し、第1段階としてそのユーザは、その供給者よりその第2ファイルに記録されている2組のシンδροーム情報を受け取り、その2組のシンδροーム情報に基づいて検査対象の一列のヌクレオチドの配列の内のその供給者の一列のヌクレオチドの配列との相違部を特定し、この相違部の配列の復元ができない場合に、第2段階としてそのユーザはその供給者よりその第1ファイルに記録されているそのテキストデータ、又はその数値データの内のその配列の復元ができない部分の情報の提供を要求することが望ましい。

【 0 0 4 4 】

このようにそのユーザが最初は、希望するヌクレオチドの配列情報のシンδροーム情報のみを購入する場合には、そのデータ量の小さいシンδροーム情報はその通信回線を介して短時間で受信することができる。そして、シンδροーム情報だけで検査対象の配列のエラーコードの特定、及び復元ができる場合には、それ

以上の配列情報を購入する必要が無い。一方、エラーコードが多く存在し、シンδροーム情報のみでは全部の正確なデータが復元できない場合には、復元できない部分のテキストデータのみを購入することによって、通信回線を介して必要な配列情報を短時間に購入できる。従って、通信回線として、携帯電話システムのような比較的低速の通信回線も使用できる。

【 0 0 4 5 】

次に、本発明のアミノ酸の配列情報の記録方法は、一列のアミノ酸の配列情報の記録方法であって、その一列のアミノ酸の配列に対応するテキストデータよりも少ないデータ量で、その一列のアミノ酸の配列に関する情報を記録するものである。

斯かる本発明によれば、その一列のアミノ酸は、例えば或るタンパク質を構成するアミノ酸の配列の少なくとも一部である。そのアミノ酸の配列が、そのテキストデータ以外のより少ないデータ量のファイルとして記録される。従って、記録媒体として、小容量でも通常のコンピュータで手軽に再生できる媒体を使用できると共に、通信回線を介して送信する際の時間を短縮できる。

【 0 0 4 6 】

本発明において、その一列のアミノ酸は、一つのタンパク質を構成する1本のアミノ酸の鎖の全部又は一部である場合に、一例としてその一列のアミノ酸の配列に対応するテキストデータが、20種類のアミノ酸に対して互いに異なる6ビット以下のデータを割り当てることによって変換される。テキストデータ形式で一文字表記 (1-Letter Code) を行うものとする、20種類のアミノ酸は、それぞれ8ビットのアスキーコード (ASCII CODE)、文字では例えばA, C, E等で表されるため、各アミノ酸を6ビット以下のデータで表すことによって、データ量を減らすことができる。

【 0 0 4 7 】

なお、一つのアミノ酸の種類は一列の3個のヌクレオチドの配列、即ち一つの遺伝子コドン (codon) によって決定される。これに関して、上記のヌクレオチドの配列情報の記録方法において、各ヌクレオチドを2ビットのデータで表した場合に、1つの遺伝子コドンは6ビットのデータで表される。そこで、この各遺

伝子コドンの 6 ビットのデータを、対応するアミノ酸の 6 ビットのデータとみなしてもよい。この場合には、所定のアミノ酸を表すデータが複数存在する、即ちコードの縮重 (degeneracy) が生じるため、一例として各アミノ酸のデータの中で最も小さいデータをそのアミノ酸に割り当てるようにしてもよい。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。また、20 種類のアミノ酸は、最も少ないデータ量では、5 ビットのデータで表すことができる。

【 0 0 4 8 】

また、本発明において、より具体的な第 1 の記録方法は、その一列のアミノ酸の配列に関する情報を、その配列を表すテキストデータの数学的な要約値 (message digest) で表すものである。例えば所定のハッシュ関数を用いてそのテキストデータの要約値を求め、この要約値をインターネット上で公開することによって、そのテキストデータを公開することなく、その配列を最先に解読したことを主張 (証明) できる。更にそのテキストデータを購入したユーザが、購入したデータの要約値を求め、この要約値を公開されている要約値と比較することによって、購入したデータの同一性を確認できる。

【 0 0 4 9 】

この場合、その一列のアミノ酸は 25 個以上のアミノ酸の配列であるときに、その一列のアミノ酸の配列に関する情報を 16 ビット以上で 192 ビット以下の長さの数学的な要約値で表すことが望ましい。タンパク質の種類は、仮想的なものも含めて 10 万 ($= 10^5$) 種類程度と言われており、次の関係が成立している。

【 0 0 5 0 】

$$10^5 \doteq 2^{16.6} \quad \dots (6)$$

従って、例えばアミノ酸の配列の個数も識別データに用いるものとする、16 ビット以上の要約値を用いることによって、ほぼ全てのタンパク質を識別することができる。また、25 個以上のアミノ酸の配列のテキストデータは、一文字表記でも 200 ビット以上になるため、192 ビット以下の要約値のデータ量はテキストデータのデータ量よりも少なくなる。

【 0 0 5 1 】

また、その数学的な要約値は、その一列のアミノ酸の配列に対応するテキストデータに例えばMD 5 ハッシュ関数（要約値は1 2 8 ビット）、又はSHS（Secure Hash Standard）ハッシュ関数（要約値は1 6 0 ビット）の演算を施して得られる。この場合、要約値が必要以上に長くない点ではMD 5 ハッシュ関数が望ましい。但し、タンパク質を構成するアミノ酸の配列の数は2 0 個～1 0 0 0 個程度であり、要約値から元のテキストデータが推定される恐れがある。そこで、アミノ酸の配列の要約値を算出する場合で、かつ元のテキストデータの秘匿性を高めたい場合には、より複雑な演算を行って得られる要約値も長くなるSHSハッシュ関数を使用することが望ましい。

【 0 0 5 2 】

また、本発明において、より具体的な第2の記録方法は、その一列のアミノ酸の配列に対応するテキストデータを、そのアミノ酸の配列方向に複数行で、かつその配列方向に交差する非配列方向に複数列の部分テキストデータに分割し、その部分テキストデータを、それぞれ複数種類のアミノ酸に対して互いに異なる8ビット以下の数値データを割り当てることによって変換データに変換し、複数行のその変換データに各行毎にその非配列方向に第1の演算を施して第1組のシンδροーム情報を求めると共に、複数列のその変換データに各列毎にその配列方向に第2の演算を施して第2組のシンδροーム情報を求め、その第1組及び第2組のシンδροーム情報でその一列のアミノ酸の配列を表すものである。

【 0 0 5 3 】

本発明は、予めそのテキストデータ（一文字表記とする）を一列の数値データに変換した後に、その数値データを複数行で複数列の変換データに分割することと実質的に等価である。本発明によれば、例えばアミノ酸の配列に対応するテキストデータを配列方向にN個（ $i = 1 \sim N$ ）で、非配列方向にM個（ $j = 1 \sim M$ ）の部分テキストデータ $T(i, j)$ に分割し、各部分テキストデータ $T(i, j)$ が4個分のアミノ酸のテキストデータを含むものとする、元のテキストデータのデータ量 DT_2 は、以下のようになる。

【 0 0 5 4 】

$$DT2 = 4 \cdot N \cdot M \text{ (バイト)} \quad \dots (7)$$

更に、その部分テキストデータ $T(i, j)$ をそのまま変換データ $A(i, j)$ とみなすと、変換データ $A(i, j)$ はそれぞれ 32 ビットの数値データとなり、シンδροーム情報も 32 ビットのデータとなる。また、非配列方向のシンδροーム情報が 2 列あるとすると、シンδροーム情報のデータ量 $DS2$ は、以下のようになる。

【0055】

$$\begin{aligned} DS2 &= 32 (2 \cdot N + M) \text{ (ビット)} \\ &= 4 (2 \cdot N + M) \text{ (バイト)} \quad \dots (8) \end{aligned}$$

従って、仮に $N = 16$, $M = 16$ とすると、(7) 式及び (8) 式よりデータ量 $DT2$, $DS2$ は以下のようになる。

$$DT2 = 1024 \text{ (バイト)} \quad \dots (9)$$

$$DS2 = 192 \text{ (バイト)} \doteq DT2 / 5.3 \quad \dots (10)$$

従って、シンδροーム情報のデータ量は、元のテキストデータのデータ量のほぼ $1/5$ 程度に圧縮できる。個々のタンパク質の配列のテキストデータのデータ量は 1 k バイト程度以下であるが、例えば 10000 種類程度のタンパク質のテキストデータをまとめると 10 M バイト程度になる。この際にシンδροーム情報を用いることによって、通信回線を介して短時間で近似的な情報を送信することができる。また、シンδροーム情報を比較することによって、標準試料のアミノ酸の配列と検査対象のアミノ酸の配列との相違部（エラーコード）を効率的に検出することができる。更に、エラーコードが各行、又は各列に 1 つの変換データのみであるときには、それに対応する正確な配列を復元できる。

【0056】

特に、複数行のその変換データの各行の変換データをそれぞれその非配列方向に交互に第 1 群の変換データ及び第 2 群の変換データに分けたとき、その第 1 の演算は、所定の整数 K を用いてその第 1 群の変換データ、及びその第 2 群の変換データのそれぞれの法 K のもとの和を求める演算であり、その第 2 の演算は、複数列のその変換データの各列の変換データに対する法 K のもとの和を求める演算である場合には、2 列に跨るような長い配列の相違（バーストエラー）であって

も正確に検出、及び復元を行うことができる。

【 0 0 5 7 】

次に、本発明の一系列のアミノ酸の配列情報の記録装置は、一つのタンパク質の少なくとも一部に含まれる一系列のアミノ酸の配列情報をテキストデータとして第 1 ファイルに記録する第 1 記録手段と、その第 1 ファイルのテキストデータよりも少ないデータ量で、その一系列のアミノ酸の配列の情報を表し、この配列の情報を第 2 ファイルに記録する第 2 記録手段とを有するものである。この発明によって、本発明のアミノ酸の配列情報の記録方法を実施することができる。

【 0 0 5 8 】

この場合、その第 2 記録手段は、その一系列のアミノ酸の配列を、この配列を表すテキストデータの数学的な要約値で表すことが望ましい。

また、本発明の一系列のアミノ酸の配列情報の供給方法は、その一系列のアミノ酸の配列に対応するテキストデータ、又は複数種類のアミノ酸に対して互いに異なる 8 ビット以下の数値データを割り当てることによってそのテキストデータを変換して得られる数値データを保持する供給者が、その一系列のアミノ酸の配列の長さの情報、及びその配列を表すテキストデータ又はその数値データの数学的な要約値の情報を通信回線を介して閲覧可能な状態にしておき、その通信回線を介してその配列の長さの情報及びその数学的な要約値の情報を閲覧したユーザより、そのテキストデータ又はその数値データの少なくとも一部の情報に対する取得要求がその供給者に届いた後に、その供給者がそのユーザにそのテキストデータ又はその数値データの少なくとも一部の情報を供給するものである。

【 0 0 5 9 】

この供給方法は、上記の本発明のアミノ酸の配列情報の記録方法を、その配列情報を供給（販売）する際のビジネスモデルに適用したものである。即ち、本発明のビジネスモデルでは、或る新規のタンパク質のアミノ酸の配列を最初に決定した供給者は、その配列のテキストデータ（又はこれを変換した数値データ）よりハッシュ関数によって要約値（message digest）を算出し、この要約値を例えばインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、最初にそのタンパク質の配列を決定したこ

とを主張（証明）できる。更に、ユーザが同じ配列情報を異なる供給者から誤って購入することも防止でき、競合メーカは重複投資を避けることができる。

【 0 0 6 0 】

また、或るユーザが、その供給者よりそのタンパク質の配列情報を購入した後、購入した配列情報よりそのハッシュ関数によって要約値を算出し、その配列の長さも求める。そして、この配列の長さ、及び要約値をインターネット上で公開されている値と比較することによって、購入した配列情報が正確なものであるかどうかを極めて高い確率で確認できる。

【 0 0 6 1 】

この場合、その一列のアミノ酸が 2 5 個以上のアミノ酸の配列であるときに、その数学的な要約値は、1 6 ビット以上で 1 9 2 ビット以下のデータであることが望ましい。

次に、本発明の第 1 の要約値の計算方法は、一つ又は複数のファイルに記録されたデータの要約値を計算するための要約値の計算方法であって、その一つ又は複数のファイルに記録されたデータの中で所定のコードを無視して要約値を計算するものである。

【 0 0 6 2 】

本発明によれば、例えばヌクレオチドの配列を表すテキストデータの要約値を計算する場合に、その配列を見やすくするために所々にスペース、改行、及びそれまでのヌクレオチドの数を示す数字等が付加されていても、これらの付加されたコードを無視することによって、本来のヌクレオチドの配列に対応する要約値を計算することができる。

【 0 0 6 3 】

この場合、その無視する所定のコードの別の例は、同一又は互いに異なる 2 組のコード、及びこれら 2 組のコードに挟まれたデータである。即ち、例えばいわゆるコメント文を要約値の計算対象から除去することによって、コメント文の内容を任意に記載しても、本来のヌクレオチドの配列に対応する要約値を計算することができる。

【 0 0 6 4 】

この場合、そのテキストデータは、その所定のコードの他に 25 個以上のヌクレオチドの配列に関するデータを含むものとする、一例として、その要約値は、40 ビット以上で 192 ビット以下のデジタルデータである。また、その要約値を計算するための関数としては、MD5 ハッシュ関数（要約値は 128 ビット）、又は SHA ハッシュ関数（要約値は 160 ビット）等を使用することができる。

【 0 0 6 5 】

また、本発明において、その一つ又は複数のファイルから 1 文字分のコードデータを読み出す毎に、この読み出されたコードデータがその所定のコードであるときには、この読み出されたコードデータを無視して、次の 1 文字分のコードデータの読み出しを行い、この読み出しによって得られたその所定のコード以外のコードデータが予め定められた個数になるか、又は読み出すべきデータがなくなったときに、要約値の計算を行うようにしてもよい。

【 0 0 6 6 】

このように部分的に読み出されたデータ毎にその所定のコードを無視して順次要約値の計算を行うことによって、例えば最初にその一つ又は複数のファイルからその所定のコードを取り出した新たなファイルを作成して、この新たなファイルの要約値を計算する方法と比べて、記憶装置の容量がほぼ 1 / 2 程度で済む利点がある。

【 0 0 6 7 】

次に、本発明の第 2 の要約値の計算方法は、一連のテキストデータの要約値を計算するための要約値の計算方法であって、その一連のテキストデータを先頭から順に所定個数ずつのコードデータを含む複数の部分テキストデータと、その所定個数よりも少ない個数のコードデータを含む端数のテキストデータとに分割し、その複数の部分テキストデータ、及び端数のテキストデータをそれぞれ分割する順序を含むデータとともに互いに異なる複数のファイルに記録し、この複数のファイルに記録されたテキストデータからその分割の順序に従って順次要約値を計算するものである。

【 0 0 6 8 】

斯かる本発明によれば、例えば人間のゲノム情報のような膨大な量のテキストデータの要約値を求める場合に、そのテキストデータを複数のファイルに分割して記録しておき、分割されたファイルのデータに順次演算処理を施すことができる。従って、CD-ROMやフレキシブルディスクのように比較的容量の少ない記録媒体を用いる場合にも、その膨大な量のテキストデータの要約値を容易に、かつ正確に計算できる。

【 0 0 6 9 】

この場合、そのその所定個数ずつのコードデータ、及びその所定個数よりも少ない個数のコードデータからは、所定のコードデータ（例えば数字コード、スペースコード、改行コード等）を除外してもよい。

また、その所定個数は、要約値を計算する際のデータ量の単位に応じて定めることが望ましい。例えばMD5ハッシュ関数は、512ビット（64バイト）のデータ単位で要約値を計算するため、一つのコードデータが8ビット（1バイト）であるとする、その所定個数は、64の整数倍に設定することによって、各部分テキストデータ毎の要約値の計算が容易になる。

【 0 0 7 0 】

【発明の実施の形態】

以下、本発明の実施の形態の一例につき図面を参照して説明する。本例は、所定のDNA（デオキシリボ核酸：deoxyribonucleic acid）のヌクレオチドの配列情報をコンピュータシステムで処理する場合に、本発明を適用したものである。

【 0 0 7 1 】

図1は、本例のコンピュータシステム2Aの概略構成を示し、この図1において、コンピュータシステム2Aの中心は、CPU（中央演算処理ユニット）、RAM、ROM等のメモリ、及びハードディスク装置等の記憶装置等からなる情報処理装置10である。情報処理装置10には、ビデオRAM（VRAM）11を介してCRTディスプレイよりなる表示装置12が接続されると共に、I/Oユニット（入出力装置）14を介して、記録可能なCD-Recordableディスク（以下、「CD-R」と言う）16に対するデータの書き込み、及びCD-RやCD

ーROMからのデータの読み込みを行うことができるCD-R/RWドライブ15が接続されている。情報処理装置10には、I/Oユニット14を介して更に大容量の記憶装置としての記憶容量が数10Gバイト程度の磁気ディスク装置17が接続されている。

【0072】

本例の情報処理装置10中のハードディスク装置には、予めCD-R/RWドライブ15を介してオペレーティングシステム、及び後述のようにDNAの配列情報を処理するためのアプリケーション・プログラムがインストールされている。また、CD-R16が本発明の記録媒体に対応しているが、記録媒体としては、CD-RやCD-ROMの他に、フラッシュROM、フレキシブルディスク、光磁気ディスク(MO)、デジタルビデオディスク(DVD)、又はハードディスク装置(例えばインターネットを介して接続できるサーバに備えられたもの)等を使用することができる。

【0073】

情報処理装置10には更に、文字情報の入力装置としてのキーボード13、ポインティング・デバイス(入力装置)としての光学式のマウス204、及びルータ(又はモデム等でもよい)よりなる通信制御ユニット18が接続されている。マウス204は、表示装置12の表示画面上のカーソルの位置を指定する信号を発生する変位信号発生部207、選択すべき情報を指定する信号や各種コマンド等を発生するための左スイッチ204a及び右スイッチ204b(信号発生装置)を備えている。情報処理装置10、VRAM11、表示装置12、キーボード13、マウス204、I/Oユニット14、CD-R/RWドライブ15、磁気ディスク装置17、及び通信制御ユニット18等よりコンピュータシステム2Aが構成されている。オペレーティングシステムとして本例ではWindows(Microsoft Corporationの登録商標)を使用している。なお、オペレーティングシステムとして、それ以外のUNIX(X/Openの登録商標)、OS/2(IBM Corporatinの登録商標)、MacOS(Apple Computerの登録商標)、又はLinux(Linus Torvaldsの商標又は登録商標)等を使用する場合にも本発明が適用できることは言うまでもない。

【 0 0 7 4 】

そして、コンピュータシステム 2 A（情報処理装置 1 0）は、通信制御ユニット 1 8 を介して一般電話回線よりなる通信ネットワーク 1 に接続され、通信ネットワーク 1 には各種コンテンツのプロバイダ 3、及び別のコンピュータシステム 2 B、及び不図示の多くのサーバやコンピュータシステムが接続されている。また、本例のコンピュータシステム 2 A、2 B 及びプロバイダ 3 は、通信ネットワーク 1 を介するインターネットによって相互に接続されている。この場合、コンピュータシステム 2 A の所有者が DNA 情報の供給者（販売者）であり、コンピュータシステム 2 B の所有者がその DNA 情報のユーザ（購入者）である。そして、後者のコンピュータシステム 2 B には、予め前者のコンピュータシステム 2 A と同様の DNA の配列情報を処理するためのアプリケーション・プログラムがインストールされている。

【 0 0 7 5 】

さて、本例のコンピュータシステム 2 A の情報処理装置 1 0 には、I/O ユニット 1 4 を介して DNA 中の一列のヌクレオチドの配列（又は塩基の配列）を読み取るための配列読み取り装置としてのシーケンサー（DNA Sequencer）4 が接続されている。シーケンサー 4 は、一例としてサンガーの方法（Sanger method）によって DNA を構成する 1 対の重合体の鎖の一方の鎖のヌクレオチドの配列を読み取る。サンガーの方法は、例えば文献 1（Maxim D. Frank-Kamenetskii: Unraveling DNA (the most important molecule of life, revised and updated), translated by Lev Liapin, Chapter 6(pp.59-70)(Perseus Books,1997)）に開示されている。シーケンサー 4 は、読み取った一列のヌクレオチドの配列をテキストデータ形式で内部の大容量の記憶装置に記憶すると共に、情報処理装置 1 0 からの要求に応じて、その記憶装置中の所定のヌクレオチドの配列のテキストデータを I/O ユニット 1 4 を介して情報処理装置 1 0 に供給する。これに対して情報処理装置 1 0 は、DNA の配列情報を処理するためのアプリケーション・プログラムに基づいて以下の処理を行う。なお、シーケンサー 4 の代わりに、DNA 及び RNA（リボ核酸：ribonucleic acid）等の核酸を構成する一列のヌクレオチドの配列（又は塩基の配列）の情報のデータベースを接続してもよい。

【 0 0 7 6 】

先ず、本例の情報処理装置 1 0 の第 1 の基本的な処理動作につき説明する。情報処理装置 1 0 は、シーケンサー 4 から供給される所定の DNA のヌクレオチドの配列を示すテキストデータを磁気ディスク装置 1 7 中のマスターファイル 1 9 にそのまま記録すると共に、そのテキストデータをよりデータ量の少ない数値データに変換し、この変換後の数値データを磁気ディスク装置 1 7 中のワーキングファイル 2 0 に記録する。なお、以下の説明において、2 進数表示の数 k は $\text{bin}(k)$ で、1 6 進数表示の数 k は $\text{hex}(k)$ で表すものとする。

【 0 0 7 7 】

この場合、DNA は 4 種類のヌクレオチドより構成されており、シーケンサー 4 から供給されるテキストデータ中では、塩基としてアデニン (adenine) を含むヌクレオチド、グアニン (guanine) を含むヌクレオチド、シトシン (cytosine) を含むヌクレオチド、及びチミン (thymine) を含むヌクレオチドがそれぞれ文字 A, G, C, 及び T で表されている。そして、文字 A, G, C, 及び T には、データ上ではそれぞれ $\text{hex}(41)$, $\text{hex}(47)$, $\text{hex}(43)$, $\text{hex}(54)$ よりなる 1 バイト (8 ビット) のアスキーデータが割り当てられている。また、RNA の場合には、チミンを含むヌクレオチドの代わりにウラシル (uracil) を含むヌクレオチドが、文字 U ($\text{hex}(55)$) で表されている。従って、 n 個のヌクレオチドの配列を示すテキストデータのデータ量は n バイトとなる。なお、それらの n 個のヌクレオチドの配列は、 n 個の塩基 (アデニン、グアニン、シトシン、チミン (又はウラシル)) の配列ともみなすことができる。

【 0 0 7 8 】

本例ではそのテキストデータを、情報量を少なくすることなく最も少ないデータ量で表すために、DNA 中の 4 種類のヌクレオチドを互いに異なる 2 ビットのデータで表す。この際に、DNA においては、1 対の塩基 (アデニン及びチミン) が互いに相補的であり、別の 1 対の塩基 (グアニン及びシトシン) が互いに相補的である。そこで、相補的な塩基を含む 1 対のヌクレオチドを互いに相補的であるとして、1 対の互いに相補的なヌクレオチド、即ちアデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、互いにビット反転の関係にある 1 対のデ

ータを割り当て、別の1対の互いに相補的なヌクレオチド、即ちグアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに、互いにビット反転の関係にある別の1対のデータを割り当てる。本例ではそのデータの割り当てとして表1（変換テーブル）を用いる。なお、表1は、ヌクレオチドの配列を示すテキストデータ中の文字A，T（又はU），G，C，をそれぞれbin(00),bin(11),bin(01),bin(10)で置換することを意味している。

【0079】

《表1》

ヌクレオチド	2ビットのデータ
アデニンを含むヌクレオチド（A）	bin(00)
チミン（ウラシル）を含むヌクレオチド（T又はU）	bin(11)
グアニンを含むヌクレオチド（G）	bin(01)
シトシンを含むヌクレオチド（C）	bin(10)

【0080】

なお、本例では各ヌクレオチドを2ビットのデータで表しているが、これは各塩基を2ビットのデータで表すのと等価である。また、データの割り当ては表1には限定されず、例えばチミンを含むヌクレオチドをbin(00)、アデニンを含むヌクレオチドをbin(11)とするか、又はグアニンを含むヌクレオチドをbin(10)、シトシンを含むヌクレオチドをbin(01)としてもよい。それ以外に、アデニンを含むヌクレオチド及びチミンを含むヌクレオチドに、1対のデータbin(01),bin(10)を割り当て、グアニンを含むヌクレオチド及びシトシンを含むヌクレオチドに1対のデータbin(00),bin(11)を割り当てるようにしてもよい。また、RNAの場合には、チミンを含むヌクレオチドに割り当てられているデータをウラシルを含むヌクレオチドに割り当てて、それ以外のヌクレオチドにはDNAのヌクレオチドと同じデータを割り当てればよい。

【0081】

本例では図2に示すDNA分子5のヌクレオチドの配列情報を扱うものとする。その配列情報は、NCBI（The National Center for Biotechnology Information）より提供されているウェブサイト1（<ftp://ncbi.nlm.nih.gov/genbank>）

/genomes/bacteria/) より入手した大腸菌 (*Escherichia coli*: *E. coli*) の DNA の一列のヌクレオチドの配列の一部である。

【 0 0 8 2 】

図 2 において、DNA 分子 5 は、1 対の重合体の鎖 6 A、6 B (二重らせん) より構成され、一方の重合体の鎖 6 A は、アデニンを含むヌクレオチド 7 A、グアニンを含むヌクレオチド 7 G、シトシンを含むヌクレオチド 7 C、及びチミンを含むヌクレオチド 7 T よりなる 4 種類のヌクレオチドの配列であり、他方の重合体の鎖 6 B は、鎖 6 A に対して相補的なヌクレオチドの配列である。この際に、図 1 の情報処理装置 1 0 には一方の重合体の鎖 6 A の配列を示すテキストデータ、即ち "AGCTTTT..." の文字列のデータが供給される。それに対して、情報処理装置 1 0 は、そのテキストデータを後述のように N 行で M 列 (N, M は 2 以上の整数) のブロックに分割した後、各ブロック中の文字 A, G, C, T を表 1 の変換テーブルに基づいて順次 2 ビットのデータに変換することによって、数値データとしてのバイナリーデータ BNA (=bin(0001101111...)) を得る。そして、このバイナリーデータ BNA が図 1 の磁気ディスク装置 1 7 のワーキングファイル 2 0 に記録される。そのバイナリーデータ BNA のデータ量は、元のテキストデータの 1/4 となっている。

【 0 0 8 3 】

この場合、そのワーキングファイル 2 0 の先頭の所定数のバイトの領域に、例えばその配列が DNA 又は RNA のどちらかを示すデータ (即ち、bin(11) を文字 T 又は文字 U の何れに解釈するかを示すデータ)、ヌクレオチドの個数を示すデータ、及びその他の必要なデータを記録しておけばよい。また、そのワーキングファイル 2 0 の長さが 1 バイト (8 ビット) 単位で規定されている場合に、バイナリーデータ BNA の末尾で 1 バイトの端数のデータが生じたときには、予め定めておいたダミーデータを付加すればよい。これでもデータ量は殆ど増加しない。そして、一例としてユーザ (コンピュータシステム 2 B の所有者) から供給者 (コンピュータシステム 2 A の所有者) に対して図 2 の DNA 分子 5 の配列情報の購入希望が届いたときに、ワーキングファイル 2 0 のデータが通信ネットワーク 1 及び不図示のプロバイダを介して、電子メールの添付ファイルとしてコン

コンピュータシステム 2 B 側に供給される。この際に、そのワーキングファイル 2 0 のデータを更に圧縮ファイル（Z I P ファイル、又は L H A ファイル等）として送信してもよい。この際に、ワーキングファイル 2 0 のデータ量はもとのテキストデータのほぼ 1 / 4 であるため、元のテキストデータ（更に圧縮ファイルとした場合も同様）自体を送信する場合に比べて送信時間はほぼ 1 / 4 となり、供給者側及びユーザ側双方の通信コストが低減できる。

【 0 0 8 4 】

そして、ユーザ側で、その受信したワーキングファイル 2 0 のデータから図 2 の一方の重合体の鎖 6 A の配列のテキストデータを復元する場合には、コンピュータシステム 2 B において、ワーキングファイル 2 0 中のバイナリーデータ B N A を、表 1 を用いて文字 A, G, C, T（又は U）の何れかに順次逆変換すればよい。また、その際に例えば図 2 の他方の相補的な重合体の鎖 6 B のヌクレオチドの配列を示すテキストデータが必要になった場合には、コンピュータシステム 2 B において、図 2 に示すように、バイナリーデータ B N A のビット毎の反転操作を行って反転バイナリーデータ NOT(BNA) (=bin(1110010000...)) を得る。この反転バイナリーデータ NOT(BNA) は、他方の重合体の鎖 6 B のヌクレオチドの配列を示すテキストデータ（文字列” T C G A A A . . . ”）を表 1 に従って変換したバイナリーデータ B N B と同一である。従って、その反転バイナリーデータ NOT(BNA) を、表 1 を用いて文字 A, G, C, T（又は U）の何れかに順次逆変換するのみで、極めて高速に相補的な重合体の鎖 6 B の配列のテキストデータを得ることができる。この際に、通常のコンピュータにおいては、ビット毎の反転操作は、極めて高速に実行することができる。なお、そのビット毎の反転操作は、例えば bin(111111...) との排他的論理和演算で代用してもよい。

【 0 0 8 5 】

なお、ワーキングファイル 2 0 のデータを通信ネットワーク 1 を介してユーザ側に送信する代わりに、ワーキングファイル 2 0 の内容を C D - R / R W ドライブ 1 5 によって C D - R 1 6 に記録し、この C D - R 1 6 を郵送等によってユーザ側に供給してもよい。例えば一人の人間の全部の D N A の配列情報（ゲノム）は、テキストデータでは 3 G バイト程度になるが、これを表 1 を用いて本例の数

値データとしてのバイナリーデータに変換すると、3/4 Gバイト程度、即ち750 Mバイト程度になる。現在のCD-R、CD-ROMの容量は約650 Mバイトであるため、その750 Mバイト程度のバイナリーデータは例えば一部又は全部を圧縮ファイルとすることによって、余裕を持ってCD-R 16に記録することができる。これに対して、その750 Mバイト程度のデータを通信ネットワーク1を介して送信しようとする、現状でも送信時間がかかり過ぎる場合がある。

【0086】

また、一つのアミノ酸の種類は一系列の3個のヌクレオチドの配列、即ち一つの遺伝子コドン(codon)によって決定される。そこで、1つのアミノ酸に対応する3個のヌクレオチドをそれぞれ2ビットのデータで表したときに得られる6ビットのデータの中で、最も小さいデータでそのアミノ酸を表すものとする。具体的に、各ヌクレオチドを表1のように表した場合に、いくつかのアミノ酸について得られる6ビットの表現を以下の表2に示す。表1中で<>の中のデータがそのアミノ酸のデータとして使用される。これによって、ヌクレオチドとアミノ酸とで共通のコードを使用できる利点がある。

【0087】

《表2》

アミノ酸	遺伝子コドン	6ビットのデータ
アラニン(Ala)	G C A	<bin(011000)>
	G C G	bin(011001)
	G C C	bin(011010)
	G C U	bin(011011)
システイン(Cys)	U G C	<bin(110110)>
	U G U	bin(110111)
グルタミン酸(Glu)	G A A	<bin(010000)>
	G A G	bin(010001)
ヒスチジン(His)	C A C	<bin(100010)>
	C A U	bin(100011)

イソロイシン(Ile)	A U A	<bin(001100)>
	A U C	bin(001110)
	A U U	bin(001111)
リジン(Lys)	A A A	<bin(000000)>
	A A G	bin(000001)。

【 0 0 8 8 】

次に、本例の情報処理装置 1 0 の第 2 の基本的な処理動作につき説明する。本例では、ヌクレオチドの配列を示す膨大な量のテキストデータ（又はこれを表 1 に基づいて変換して得られる数値データより、所定のハッシュ関数を用いて数学的な要約値（message digest）を算出する。本例ではそのハッシュ関数として、ライベスト（R. Rivest）によって提案された MD 5 ハッシュ関数を使用する。MD 5 ハッシュ関数のアルゴリズムについては、ネットワークワーキンググループ及びライベストによって開設されているウェブサイト 2 (<http://www.kleinschmidt.com/edi/md5.htm>) に開示されている。或るテキストデータ（テキストファイル）にその MD 5 ハッシュ関数を施すことによって、1 2 8 ビットの要約値が得られる。通常のコンピュータでも今後は 6 4 ビットの CPU が使用されるようになると考えられるが、この場合に 1 2 8（= 2・6 4）ビットの要約値は非常に扱い易い長さである。この場合には、1 9 2（= 3・6 4）ビットの要約値も比較的扱い易いと考えられる。

【 0 0 8 9 】

また、本例では、その MD 5 ハッシュ関数のプログラムとして、そのウェブサイト 2 において公開されている、RSA データセキュリティー社（RSA Data Security Inc.）によって開発されたプログラムを使用した。

その要約値の使用方法の一例として、DNA の配列情報の供給者（情報処理装置 1 0）は、所定の生物の DNA のヌクレオチドの配列を読み取り、これに対応するテキストデータより、上記のハッシュ関数を用いて要約値を算出し、この要約値をその生物の名称、及び DNA の位置を示す情報と共にインターネット上で閲覧可能にする。これによって、その供給者は、そのテキストデータ自体を公開することなく、その生物の DNA の配列情報を最先に解読したことを主張できる

と考えられる。その後、或るユーザからのその配列情報の購入希望が来たときに、その供給者は、そのヌクレオチドの配列のテキストデータを表 1 を用いてバイナリーデータに変換し、このバイナリーデータを例えば通信ネットワーク 1 を介して電子メールの形でそのユーザに送信する。これに対してユーザ側では、そのバイナリーデータを表 1 を用いてテキストデータに逆変換し、この逆変換されたテキストデータに上記のハッシュ関数を施して要約値を求める。

【 0 0 9 0 】

そして、この要約値とその供給者によって公開されている要約値とが等しいときには、購入した配列情報が、供給の保持している配列情報と等しいことが極めて高い確率で保証される。更に、ユーザ側では、複数の供給者が公開している要約値を比較することによって、同じ配列情報を異なる複数の供給者から重複して購入することを防止することができる。これらの際に、ヌクレオチドの配列の長さ、及び先端部や末尾の一部の短い配列の比較を行うことによって、その配列情報の同一性を高めることができる。

【 0 0 9 1 】

また、例えば所定の生物について複数の研究者が並行して研究を行っている場合に、第 1 の研究者が保持しているヌクレオチドの配列と第 2 の研究者が保持しているヌクレオチドの配列との同一性を保証する必要がある。この際に、研究対象とする DNA のヌクレオチドの配列数が例えば 1 億個程度とすると、その配列のテキストデータは 1 0 0 M バイト程度になる。このような長い 2 つのテキストデータに対して 1 文字ずつの比較によって、同一性を確認するのは容易ではない。これに対して本例では、先ず第 1 の研究者側で、テキストデータの長さ、及びハッシュ関数による要約値を算出し、これを電子メール等で第 2 研究者側に送信する。これに対して、第 2 の研究者側でも自分のテキストデータの長さ及びハッシュ関数による要約値を算出し、これらの値を第 1 の研究者から送信された値と比較することによって、2 つの膨大な長さのテキストデータの同一性を容易に高い確率で保証できる。この際にも、更に例えばヌクレオチドの配列の先端部及び末尾の所定長さの配列同士を比較することによって、その同一性を高めることができる。

【 0 0 9 2 】

なお、ハッシュ関数としては、例えば文献 2 (FIPS Publication 180, 1993) で開示されているように、N B S (National Bureau of Standards) によって提案された S H S (Secure Hash Standard) ハッシュ関数を使用してもよい。S H S ハッシュ関数は、M D 5 ハッシュ関数よりも複雑な演算を行うと共に、1 6 0 ビットの要約値が得られる。これに関して、例えばタンパク質を構成するアミノ酸の配列数は 2 0 個～1 0 0 0 個程度であり、特に一文字表記を使用する際にはそれに対応するテキストデータも 2 0 バイト～1 k バイト程度に短くなるため、要約値から元のテキストデータが推定し易いと考えられる。そこで、アミノ酸の配列情報の要約値を求める際には、S H S ハッシュ関数を使用する方が望ましいことがある。

【 0 0 9 3 】

また、例えばヌクレオチドの配列を示す 2 つの膨大な長さのテキストデータの同一性を確認するために、ハッシュ関数の要約値を算出するような場合には、それ程複雑な計算を繰り返して行う必要は無いと考えられる。そこで、このような用途では、例えば文献 3 (R. L. Rivest: "The MD4 message digest algorithm", Lecture Notes in Computer Science, 537, 303-311 (1991)) で開示されている M D 4 ハッシュ関数を使用してもよいと考えられる。また、そのように単に同一性を確認する用途では、要約値の長さも 4 0 ビット～1 2 8 ビット程度でよい場合がある。

【 0 0 9 4 】

次に、本例の D N A 情報の供給者 (コンピュータシステム 2 A) と、ユーザ (コンピュータシステム 2 B) との間で D N A の配列情報を受け渡す際のビジネスモデルの一例につき図 3 ～図 6 のフローチャートを参照して詳細に説明する。まず、D N A 情報の供給者側では、図 3 のステップ 1 0 1 において、シーケンサー 4 を使用して標準となる試料 (標準試料 E とする) の D N A 中の一方の系列のヌクレオチドの配列を読み取り、読み取った配列を表すテキストデータ T X 1 を情報処理装置 1 0 に供給する。本例では、その標準試料 E を大腸菌として、そのテキストデータ T X 1 として、図 7 に示すように、上記のウェブサイト 1 から入手

した大腸菌のDNAの配列情報の内の、最初から2048個までのヌクレオチドの配列を示すテキストデータを使用する。

【0095】

標準試料EのDNA配列は配列番号1に示されている。図7のテキストデータは、配列番号1の配列から数字データを除いて、a, g, c, tの文字をそれぞれA, G, C, Tで置き換えたものに相当する。

次のステップ102において、情報処理装置10は、供給されたテキストデータTX1に上記のMD5ハッシュ関数を施して128ビットの要約値AB1を求めると共に、そのヌクレオチドの配列の数NA1、及び先頭と末尾との8個ずつのヌクレオチドの配列ST1, SB1を求める。テキストデータTX1に対する具体的な値は下記の通りである。

【0096】

AB1 = hex(849339ac244cde42b5346ab5989aab61) ... (11)

NA1 = 2048

ST1 = AGCTTTTC, SB1 = CGCGAAGG

次のステップ103において、情報処理装置10は、テキストデータTX1を逆方向に並べ替えたテキストデータTXR1 (=GGAAGC...TTTCGA)を求め、このテキストデータTXR1のMD5ハッシュ関数による要約値ABR1、及びこのテキストデータTXR1の先頭と末尾との8個ずつのヌクレオチドの配列STR1, SBR1を求める。配列STR1, SBR1は、上記の配列SB1, ST1をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。

【0097】

ABR1 = hex(4eb1feae30f522642b912ce3ea09652b) ... (12)

STR1 = GGAAGCGC, SBR1 = CTTTTTCGA

次のステップ104において、情報処理装置10は、標準試料Eの名前の情報(試料を特定する情報)、配列の数NA1、テキストデータTX1、配列ST1, SB1、要約値AB1、逆方向の配列STR1, SBR1、及び逆方向の要約値ABR1を磁気ディスク装置17のマスターファイル19に記録する。この際

に、マスターファイル 19 を複数のファイルとして、テキストデータ TX1 と、それ以外のデータとを別のファイルに記録してもよい。また、テキストデータ TX1 が例えば 100M バイト程度以上になる場合には、テキストデータ TX1 を複数のマスターファイルに分割して記録してもよい。

【0098】

次のステップ 105 において、情報処理装置 10 は、図 7 に示すように、標準試料 E のテキストデータ TX1 を配列方向（ヌクレオチドの配列方向）に N 行で、その配列方向に直交する方向（以下、「非配列方向」という）に M 列の 16 文字の長さの部分テキストデータ $T(i, j)$ ($i = 1 \sim N$, $j = 1 \sim M$) に分割する。なお、N, M はそれぞれ 2 以上の任意の整数であり、(1) 式、(2) 式を用いて既に説明したように、テキストデータ TX1 が 100k バイト程度（又はこの整数倍）であるときに、このテキストデータ TX1 に対して $1/100$ 程度のデータ量のシンドローム情報を得たい場合には、例えば N の値が 64、M の値が 128 に設定される。以下では説明を簡単にするために、テキストデータ TX1 を 4 行で、かつ 32 列に分割した場合を想定する。即ち、 $N = 4$, $M = 32$ とする。この場合、本例では端数は生じないが、例えば図 7 において、最後の部分テキストデータ $T(4, 32)$ 中の文字が 16 個より少ない場合には、足りない部分には予め定めた文字（例えば文字 A）をダミーデータとして付加すればよい。また、部分テキストデータ $T(i, j)$ の長さは、16 文字以外の任意の長さでよいが、処理速度を高めるためには、8 文字の倍数が効率的である。

【0099】

更に、情報処理装置 10 は、図 7 の各部分テキストデータ $T(i, j)$ を表 1（変換テーブル）に基づいてそれぞれ 32 ビットのバイナリーデータ（数値データ）よりなる変換データ $A(i, j)$ に変換する。この結果、図 8 に示す 4 行で、32 列の変換データ $A(i, j)$ （16 進数表示）が得られる。また、変換データ $A(i, j)$ を対応するヌクレオチドの配列方向に連続して配列したときの集合体（数値データ）をバイナリーデータ BN1 とする。このバイナリーデータ BN1 は、図 2 の一方のバイナリーデータ BNA と同じものであるが、図 2 のバイナリーデータ BNA は 2 進数表示されており、図 8 のバイナリーデータ（変換

データ $A(i, j)$ は 16 進数表示されている。この場合、各部分テキストデータ $T(i, j)$ の長さは 16 バイト (= 128 ビット) であるため、図 7 の全体のテキストデータ $TX1$ に対して、図 8 の全体のバイナリーデータ $BN1$ のデータ量は $1/4$ に減少している。なお、図 7 の部分テキストデータ $T(i, j)$ と図 8 の変換データ $A(i, j)$ とは等価であるため、上記の方法の代わりに、元のテキストデータ $TX1$ を表 1 に基づいてバイナリーデータ $BN1$ に変換した後、このバイナリーデータ $BN1$ を N 行で、 M 列の変換データ $A(i, j)$ に分割してもよい。

【0100】

次のステップ 106 において、情報処理装置 10 は、図 8 の全部の変換データ $A(i, j)$ の内で、各列の変換データ $A(i, j)$ の配列方向に対する法 2^{32} ($\text{mod } 2^{32}$) のもとでの和、即ち配列方向のシンδροーム (syndrome) $C(j)$ ($j = 1 \sim 32$) を計算する。 $C(j)$ は以下のように表すことができる。

$$C(j) = A(1, j) + A(2, j) + \dots + A(4, j) (\text{mod } 2^{32}) \quad \dots (13)$$

更に、情報処理装置 10 は、各行の変換データ $A(i, j)$ ($i = 1 \sim 4$) の内で奇数番目の変換データ $A(i, 2j' - 1)$ ($j' = 1 \sim 16$) の非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データ $A(i, 2j')$ の非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向のシンδροーム $B1(i)$, $B2(i)$ ($i = 1 \sim 4$) を次式より計算する。

【0101】

$$B1(i) = A(i, 1) + A(i, 3) + \dots + A(i, 31) (\text{mod } 2^{32}) \quad \dots (14)$$

$$B2(i) = A(i, 2) + A(i, 4) + \dots + A(i, 32) (\text{mod } 2^{32}) \quad \dots (15)$$

変換データ $A(i, j)$ に対する実際の計算結果が、図 8 のシンδροーム $C(j)$, $B1(i)$, $B2(i)$ として表示されている。

また、図 9 は、図 8 の標準試料 E のデータ中からシンδροーム $C(j)$, $B1(i)$, $B2(i)$ だけを取り出して表示したものである。この例においては、シンδροーム $C(j)$, $B1(i)$, $B2(i)$ はそれぞれ 32 ビット (4 バイト) であるため、全部のシンδροームのデータ量は、160 (= 4 · 40) バイトとなる。従って、全部のシンδροームのデータ量は、図 7 の全体のテキストデ

ータTX1（2048バイト）に対してほぼ1／13に減少しており、図8の全体のバイナリーデータBN1に対してもほぼ1／3に減少している。

【0102】

次に図4のステップ107において、情報処理装置10は、標準試料Eの名前の情報、配列の数NA1、バイナリーデータBN1、シンドロームC（j），B1（i），B2（i）を磁気ディスク装置17のワーキングファイル20に記録する。この際に、ワーキングファイル20を複数のファイルとして、バイナリーデータBN1と、シンドロームC（j），B1（i），B2（i）とを別のファイルに記録してもよい。更に、バイナリーデータBN1と共に、ステップ102で算出した要約値AB1をワーキングファイル20に記録してもよい。

【0103】

また、バイナリーデータBN1が長いときには、バイナリーデータBN1を複数のファイルに分割して記録してもよい。更に、図7のテキストデータTX1（ひいては図8のバイナリーデータBN1）がかなり長い場合には、テキストデータTX1を例えば100kバイト程度を単位として複数のデータ群に分割し、各データ群毎にシンドロームC（j），B1（i），B2（i）を求めるようにしてもよい。

【0104】

更に、ステップ107において、DNA情報の供給者は、ワーキングファイル20に記録した情報、即ち標準試料Eの名前の情報、配列の数NA1、バイナリーデータBN1、シンドロームC（j），B1（i），B2（i）と、マスターファイル17に記録した要約値AB1，ABR1の情報とを、CD-R／RWドライブ15を介してCD-R16に記録してもよい。このCD-R16から、更に多数のCD-ROMを作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

【0105】

次の、ステップ108において、情報処理装置10は、標準試料Eの名前の情報、配列の数NA1、配列ST1，SB1、要約値AB1、逆方向の配列STR1，SBR1、及び逆方向の要約値ABR1を磁気ディスク装置17のコンテン

ツファイル 21 に記録する。図 7 のテキストデータ TX1 が仮に 100M バイト程度の膨大なものであっても、コンテンツファイル 21 に記録されるデータは 500 バイト程度の僅かなものである。更に、情報処理装置 10 は、コンテンツファイル 21 中の情報を通信ネットワーク 1 を介してコンテンツのプロバイダ 3 に送信する。これによって、コンテンツファイル 21 中の情報はプロバイダ 3 のサーバ内の閲覧可能なコンテンツファイル 31 に記録されて、第 3 者がインターネットを介して自由に閲覧できるようになる。

【0106】

次のステップ 109 において、DNA 情報の供給者は、ユーザから購入要求が来るのを待つ状態となる。そして、(a) ユーザから標準試料 E に対する簡易データの要求があったときには、ステップ 110 に移行して、情報処理装置 10 は、磁気ディスク装置 17 のワーキングファイル 20 中のシンドローム C (j) , B1 (i) , B2 (i) の情報を例えば電子メールの添付ファイルとしてそのユーザに送信する。一方、ステップ 109 において、(b) ユーザから完全データの要求があったときには、ステップ 111 に移行して、情報処理装置 10 は、ワーキングファイル 20 中のバイナリーデータ BN1 を ZIP ファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値 AB1 を同時に送信してもよい。本例によれば、簡易データ (シンドローム) はデータ量が少ないために短時間で送信することができる。また、完全データ (バイナリーデータ BN1) でも元のテキストデータに比べて 1/4 のデータ量であるため、比較的短時間で送信することができる。

【0107】

また、ステップ 109 において、ユーザは、必要に応じて部分データ、即ち図 8 の全部の変換データ A (i, j) の内の所望のデータ、例えば 2 つの変換データ A (4, 16) 及び A (1, 17) のみをその供給者から購入するようにしてもよい。これによって、必要な正確なデータのみを短時間で入手することができる。

【0108】

次に、DNA情報のユーザ（図1のコンピュータシステム2Bの所有者とする）側では、図5のステップ121において、図1の通信ネットワーク1（インターネット）を介してプロバイダ3のサーバ内のコンテンツファイル31の内容を閲覧し、その中からステップ108で供給者から送信された情報、即ち標準試料Eの名前の情報、ヌクレオチドの配列の数NA1、配列ST1、SB1、要約値AB1、逆方向の配列STR1、SBR1、及び逆方向の要約値ABR1を読み取り、読み取った情報をコンピュータシステム2B内の記憶装置の一時ファイルに記録する。

【0109】

次の、ステップ122において、そのユーザは、不図示のDNAのシーケンサーを用いて、標準試料Eと同じ種類で検査対象の試料FのDNA中の一方の系列のヌクレオチドの配列を読み取り、読み取られた配列を示すテキストデータTX2をコンピュータシステム2B内の情報処理装置に取り込む。その検査対象の試料Fとは、例えば突然変異を起こしていると思われる大腸菌であり、そのテキストデータTX2は、標準試料EのテキストデータTX1と同様に最初から2048個までのヌクレオチドの配列を示すものとする。

【0110】

試料FのDNA配列は配列番号2に示されている。後述の図10のテキストデータは、配列番号2の配列から数字データを除いて、a, g, c, tの文字をそれぞれA, G, C, Tで置き換えたものに相当する。

図10は、その試料FのDNAのヌクレオチドの配列に対応するテキストデータTX2を示し、この図10の配列の内のアンダーラインを付した部分のみが、図7の標準試料Eの配列と異なっている。即ち、試料Fの配列は、標準試料Eの部分テキストデータT(4, 16), T(1, 17)の部分だけが以下のように異なっている。なお、この段階では、ユーザは、試料Fの配列と標準試料Eの配列とのどの部分が相違しているのかは分からない。

【0111】

標準試料E

試料F

T(4,16)=ATTTGGACGGACGTTG → ATTTGGACATTATGGC

T(1,17)=ACGGGGTCTATACCTG → GGCCAACTTATACCTG

そして、ユーザのコンピュータシステム 2 B 側の情報処理装置においても、DNA の配列情報を処理するためのアプリケーション・プログラムが起動されている。そして、その情報処理装置は、ステップ 1 2 3 において、読み取られたテキストデータ TX 2 に上記の MD 5 ハッシュ関数を施して 1 2 8 ビットの要約値 AB 2 を求めると共に、そのヌクレオチドの配列の数 NA 2、及び先頭と末尾との 8 個ずつのヌクレオチドの配列 ST 2、SB 2 を求め、これらを内部の記憶装置の第 1 データファイルに記録する。テキストデータ TX 2 (図 1 0) に対する具体的な値は下記の通りである。

【 0 1 1 2 】

AB 2 = hex(1457b51222a83c3222e87cb4d4e63305) ... (1 6)

NA 2 = 2 0 4 8

ST 2 = AGCTTTTC, SB 2 = CGCGAAGG

次のステップ 1 2 4 において、情報処理装置は、試料 F の配列数 NA 2 と標準試料 E の配列数 NA 1 とが等しいかどうかを調べ、両者が異なっている場合には、ユーザはステップ 1 2 5 に移行して、別の DNA 情報を検索し、NA 2 と同じ配列数の DNA 情報をサーチする。本例では、ステップ 1 2 4 において、NA 2 = NA 1 であるため、動作はステップ 1 2 6 に移行して、試料 F の先頭と末尾との一部の配列 ST 2、SB 2 が、標準試料 E の配列 ST 1、SB 1 と等しいかどうか、更に試料 F の要約値 AB 2 が標準試料 E の要約値 AB 1 (ステップ 1 2 1 で一時ファイルに記録されている) と等しいかどうかを調べる。これらが共に等しい場合には、試料 F の配列と標準試料 E の配列とは非常に高い確率 (ほぼ $1 / 2^{128} \cong 1 / 10^{38}$ 程度の確率) で一致しているとみなすことができる。従って、ステップ 1 2 7 に移行して、コンピュータシステム 2 B の情報処理装置は、その第 1 データファイルに「試料 F の DNA 構造は、標準試料 E の DNA 構造と同一である。」との情報を記録する。

【 0 1 1 3 】

但し、本例では、ST 2 = ST 1、SB 2 = SB 1 が成立するが、(1 1) 式及び (1 6) 式より AB 2 ≠ AB 1 であるため、動作はステップ 1 2 6 からステ

ップ 1 2 8 に移行して、その情報処理装置は、試料 F の先頭と末尾との一部の配列 S T 2, S B 2 が、標準試料 E を逆に並べた配列の一部の配列 S T R 1, S B R 1 と等しいかどうか、更に試料 F の要約値 A B 2 が標準試料 E を逆に並べた配列の要約値 A B R 1 と等しいかどうかを調べる。これらが共に等しい場合には、試料 F の配列と標準試料 E を逆に並べた配列とは非常に高い確率で一致しているとみなすことができる。従って、ステップ 1 3 9 に移行して、コンピュータシステム 2 B の情報処理装置は、その第 1 データファイルに「試料 F の D N A 構造は、標準試料 E の D N A 構造に対して回文 (palindrome) の関係にある。」との情報を記録する。

【 0 1 1 4 】

本例では、 $S T 2 \neq S T R 2$, $S B 2 \neq S B R 2$ 、かつ (1 2) 式及び (1 6) 式より $A B 2 \neq A B R 1$ であるため、動作はステップ 1 2 8 からステップ 1 2 9 に移行して、そのユーザは、通信ネットワーク 1 (インターネット) を介して D N A 情報の供給者から上記の簡易データ、即ち標準試料 E のシンドローム C (j), B 1 (i), B 2 (i) の情報 (図 9 の情報) を購入し、購入した情報をコンピュータシステム 2 B (情報処理装置) 内の記憶装置の第 2 データファイルに記録する。

【 0 1 1 5 】

次に、図 6 のステップ 1 3 0 において、コンピュータシステム 2 B の情報処理装置は、図 1 0 に示すように、試料 F のテキストデータ T X 2 を配列方向 (ヌクレオチドの配列方向) に N 行で、非配列方向に M 列の 1 6 文字の長さの部分テキストデータ T F (i, j) ($i = 1 \sim N$, $j = 1 \sim M$) に分割する。分割数 N, M は標準試料 E の分割数と同じであり、本例では、 $N = 4$, $M = 3 2$ である。更に、情報処理装置は、図 1 0 の各部分テキストデータ T F (i, j) を表 1 (変換テーブル) に基づいてそれぞれ 3 2 ビットのバイナリーデータ (数値データ) よりなる変換データ A F (i, j) に変換する。この結果、図 1 1 に示す 4 行で、3 2 列の変換データ A F (i, j) (1 6 進数表示) が得られる。また、変換データ A F (i, j) を連続して配列した集合体 (数値データ) をバイナリーデータ B N 2 とする。

【 0 1 1 6 】

次に、情報処理装置は、ステップ 1 0 6 の動作と同様にして、図 1 1 の全部の変換データ $AF(i, j)$ の内で、各列の変換データ $AF(i, j)$ の配列方向に対する法 $2^{32} \pmod{2^{32}}$ のもとでの和、即ち配列方向のシンδροーム $CF(j)$ ($j = 1 \sim 32$) を計算する。 $CF(j)$ は、(13) 式で $A(i, j)$ を $AF(i, j)$ で置き換えた式で計算される。更に、情報処理装置は、各行の変換データ $AF(i, j)$ ($i = 1 \sim 4$) の内で奇数番目の変換データ $AF(i, 2j' - 1)$ ($j' = 1 \sim 16$) の非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データ $AF(i, 2j')$ の非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向のシンδροーム $B1F(i)$, $B2F(i)$ ($i = 1 \sim 4$) を計算する。 $B1F(i)$, $B2F(i)$ は、(14) 式、(15) 式で $A(i, j)$ を $AF(i, j)$ で置き換えた式で計算される。変換データ $AF(i, j)$ に対する実際の計算結果が、図 1 1 のシンδροーム $CF(j)$, $B1F(i)$, $B2F(i)$ として表示されている。

【 0 1 1 7 】

図 8 (標準試料 E) と図 1 1 (試料 F) とを比較すると、図 8 の変換データ $A(4, 16)$, $A(1, 17)$ に対して図 1 1 の変換データ $AF(4, 16)$, $AF(1, 17)$ の値が異なっている。従って、それに対応して図 1 1 にアンダーラインを付して示すように、図 1 1 の配列方向の 2 つのシンδροーム $CF(16)$, $CF(17)$ 、及び非配列方向の 2 つのシンδροーム $B1F(1)$, $B2F(4)$ の値が、図 8 の対応するシンδροーム $C(16)$, $C(17)$, $B1(1)$, $B2(4)$ の値と異なっている。

【 0 1 1 8 】

また、図 1 2 は、主に図 1 1 の試料 F のデータ中からシンδροーム $CF(j)$, $B1F(i)$, $B2F(i)$ だけを取り出して表示したものである。

次に、ステップ 1 3 1 において、その情報処理装置は、供給者から購入した簡易データの 1 組のシンδροーム、即ち図 8 (標準試料 E) の 1 組のシンδροーム $C(j)$, $B1(i)$, $B2(i)$ と、上記のように求めた試料 F の 1 組のシンδροーム $CF(j)$, $B1F(i)$, $B2F(i)$ とを比較して、相違するシン

ドローームをサーチする。本例では、図 1 1 のシンドローーム C F (1 6) , C F (1 7) 、及びシンドローーム B 1 F (1) , B 2 F (4) が相違するシンドローームとして特定される。この場合、配列方向の相違するシンドローーム C F (1 6) , C F (1 7) の列と、非配列方向の相違するシンドローーム B 1 F (1) , B 2 F (4) の行との交点が、標準試料 E に対して相違する変換データの位置となる。従って、図 1 1 の第 4 行で第 1 6 列の変換データ A F (4 , 1 6) 、及び第 1 行で第 1 7 列の変換データ A F (1 , 1 7) が相違する変換データとして特定される。

【 0 1 1 9 】

次のステップ 1 3 2 において、その情報処理装置は、図 1 1 の変換データ A F (i , j) 中で図 8 の変換データ A (i , j) と相違する変換データ (A F (i ' , j ') とする) は、各行、又は各列に多くとも一つかどうかを調べる。これが成立する場合には、その変換データ A F (i ' , j ') に対応する標準試料 E の変換データは、法 2^{32} のもとでの連立方程式によって容易に求めることができる。本例では、それが成立する、即ち相違する変換データは、第 1 行、第 4 行に一つずつで、かつ第 1 6 列、第 1 7 列に一つずつであるため、動作はステップ 1 3 3 に移行する。そして、その情報処理装置は、先ず変換データ A F (4 , 1 6) から標準試料 E の変換データ A (4 , 1 6) を復元するために、図 8 のシンドローーム C (1 6) 、図 1 1 のシンドローーム C F (1 6) 、及び変換データ A F (4 , 1 6) を用いて次の演算を行う。

【 0 1 2 0 】

$$\begin{aligned} A(4, 16) &= C(16) - CF(16) + AF(4, 16) \pmod{2^{32}} \\ &= \text{hex}(7c33894d) - \text{hex}(7c3373a6) + \text{hex}(3f523cd6) \pmod{2^{32}} \\ &= \text{hex}(3f52527d) \quad \dots (17) \end{aligned}$$

この結果を図 8 の変換データ A (4 , 1 6) と比較すると、復元が正確に行われていることが分かる。

【 0 1 2 1 】

続いて、情報処理装置は、変換データ A F (1 , 1 7) から標準試料 E の変換データ A (1 , 1 7) を復元するために、図 8 のシンドローーム C (1 7) 、図 1

1 のシンドローム $CF(17)$ 、及び変換データ $AF(1, 17)$ を用いて次の演算を行う。

$$\begin{aligned} A(1, 17) &= C(17) - CF(17) + AF(1, 17) \pmod{2^{32}} \\ &= \text{hex}(31b4c2ad) - \text{hex}(6661c2ad) + \text{hex}(5a0bccad) \pmod{2^{32}} \\ &= \text{hex}(255eccad) \quad \dots (18) \end{aligned}$$

この結果を図 8 の変換データ $A(1, 17)$ と比較すると、復元が正確に行われていることが分かる。また、復元された変換データ $A(4, 16)$ 、 $A(1, 17)$ が、図 12 中の試料 F のシンドローム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ の内側に表示されている。図 12 の変換データ $A(4, 16)$ 、 $A(1, 17)$ を表 1 に従って逆変換して得られる部分テキストデータは、図 7 の標準試料 E の部分テキストデータ $T(4, 16)$ 、 $T(1, 17)$ と等しいことが分かる。

【 0 1 2 2 】

次のステップ 134 において、その情報処理装置は、復元された変換データ $A(i', j')$ 、即ち $A(4, 16)$ 、 $A(1, 17)$ で、図 11 の試料 F のバイナリーデータ $BN2$ 中の対応する変換データ $AF(4, 16)$ 、 $AF(1, 17)$ を置き換えた後、この置き換えによって得られるバイナリーデータ $BN2$ を表 1 に基づいてテキストデータ $TX1'$ に逆変換する。更に情報処理装置は、そのテキストデータ $TX1'$ より MD5 ハッシュ関数を用いて 128 ビットの要約値 $AB1'$ を算出し、この要約値 $AB1'$ が標準試料 E の要約値 $AB1$ (ステップ 121 で一時ファイルに記録されている) と等しいかどうかを確認する。本例では、 $AB1' = AB1$ が成立するが、例えば図 11 の試料 F の変換データ $AF(i, j)$ 中の相違する変換データの位置や内容によっては、その相違する位置がステップ 132 で正確に検出されない可能性がある。このような場合に、 $AB1' \neq AB1$ となったときには、ステップ 135 に移行すればよい。通常は、 $AB1' = AB1$ が成立して、動作はステップ 138 に移行して、情報処理装置は、上記の第 1 データファイルに「試料 F の配列と標準試料 E の配列との内で相違する部分の位置 (i', j') 、及び相違する部分テキストデータの対」の情報を記録する。本例では、位置 (i', j') として位置 $(4, 16)$ 、 $(1, 1$

7) が、相違する部分テキストデータの対として $A(4, 16)$, $AF(4, 16)$ 及び $A(1, 17)$, $AF(1, 17)$ が記録される。

【 0 1 2 3 】

一方、ステップ 1 3 2 において、相違する変換データ $AF(i', j')$ が少なくとも一行に 2 個以上で、かつ列方向にも 2 個以上（奇数番目又は偶数番目で 2 個以上を意味する）存在する場合には、変換データの正確な復元は困難である。そこで、動作はステップ 1 3 5 に移行して、そのユーザはその DNA 情報の供給者から標準試料 E の完全データ、即ち図 8 のバイナリーデータ $BN1$ を通信ネットワーク 1（インターネット）を介して購入し、コンピュータシステム 2 B の情報処理装置は、そのバイナリーデータ $BN1$ を記憶装置の第 3 データファイルに記録する。

【 0 1 2 4 】

次のステップ 1 3 6 において、その情報処理装置は、そのバイナリーデータ $BN1$ を表 1 に基づいてテキストデータ $TX1'$ に逆変換（復元）し、そのテキストデータ $TX1'$ より MD 5 ハッシュ関数を用いて 1 2 8 ビットの要約値 $AB1'$ を算出し、この要約値 $AB1'$ が標準試料 E の要約値 $AB1$ （ステップ 1 2 1 で一時ファイルに記録されている）と等しいかどうかを確認する。通常は、 $AB1' = AB1$ が成立するが、例えば通信エラー等によって送信されたバイナリーデータ $BN1$ の中にエラーが生じている場合には、 $AB1' \neq AB1$ となる。このときには、例えば供給者に完全データの再送信を行う等の対処を行う。そして、ステップ 1 3 6 で $AB1' = AB1$ が成立するときには、ステップ 1 3 7 に移行して情報処理装置は、標準試料 E のバイナリーデータ $BN1$ 中で、試料 F の相違している変換データ $AF(i', j')$ に対応する変換データ $A(i', j')$ を求める。その後、動作はステップ 1 3 8 に移行する。

【 0 1 2 5 】

このように本例のビジネスモデルによれば、第 1 段階として標準試料 E の要約値 $AB1$ と試料 F の要約値 $AB2$ とを比較して、両者が等しいときには試料 F の DNA の構造は標準試料 E の DNA の構造と同じとみなすため、DNA 情報の供給者からそれ以上の情報を購入する必要が無い。また、第 2 段階として、標準試

料EのシンδροームC(j), B1(i), B2(i)と試料FのシンδροームCF(j), B1F(i), B2F(i)とを比較して、相違する変換データAF(i, j)の個数が少ない場合には、対応する標準試料Eの変換データA(i, j)を復元するため、膨大な完全データを購入する必要がなく、情報処理コストを低減できる。

【0126】

なお、上記のステップ135では、ユーザはDNA情報の供給者から完全データ(バイナリーデータBN1)を購入しているが、別の方法として、ステップ131で特定された相違する変換データAF(i', j')に対応する標準試料Eの変換データA(i', j')のみを購入してもよい。これによって、通信コストを低減できる。

【0127】

また、本例のシンδροームの使用方法に関して、本例では非配列方向(列方向)に2組のシンδροームB1(i), B2(i)を求めているため、図11の試料Fの変換データAF(i, j)において、連続する2列の変換データAF(i, j)、例えばAF(i, 16), AF(i, 17)(i=1~4)の全部(8個)が標準試料Eの変換データA(i, j)と相違していても、その相違する部分(エラーコード)の位置を正確に検出することができる。更に、非配列方向のシンδροームB1F(i), B2F(i)、及び変換データAF(i, 16), AF(i, 17)(i=1~4)を用いて連立方程式を解くことによって、対応する標準試料Eの変換データA(i, j)の全部を正確に復元できる。即ち、ヌクレオチドの配列方向に対して隣接する2列に跨るような比較的長いエラーコード(バーストエラー)が生じてても、本例のシンδροームによってその位置の検出、及び対応する配列の復元を行うことができる。

【0128】

また、本例のシンδροームを用いれば、SNP(一塩基変位多型: Single Nucleotide Polymorphism)のように所定の範囲内で1つのヌクレオチド(塩基)だけが異なっているようなエラーコードは、更に容易にその位置の検出、及び復元を行うことができる。そして、所定の範囲内で、即ち図11の配列中で1つ(ヌ

クレオチドの1つ分)だけ生じたエラーコードの検出、及び復元を行えば良い場合には、非配列方向のシンδροーム $B1F(i)$, $B2F(i)$ の代わりに、それらの和 ($BF(i)$ とする) を使用するのみで十分である。この場合には、図 8 の標準試料 E についても、非配列方向のシンδροーム $B1(i)$, $B2(i)$ の代わりに、それらの和 ($B(i)$ とする) を用意するのみでよい。

【 0 1 2 9 】

また、例えば図 8 (図 1 1 でも同様) において、配列方向のシンδροーム $C(j)$ を、各列で前半の 1 組の変換データと後半の 1 組の変換データとで 2 つ設け、非配列方向のシンδροーム $B1(i)$, $B2(i)$ を一つ $B(i)$ とした場合にも、上記の隣接する 2 列に跨るバーストエラーの検出及び復元を行うことができる。また、より多くのエラーコードの復元を行うためには、計算は極めて複雑になるが、シンδροーム $C(j)$, $B1(i)$, $B2(i)$ の代わりに、シンδροーム情報として例えばリードソロモンの CRC 符号 (Reed-Solomon Cyclic Redundancy Check Code: RSCRC Code) を使用してもよい。RSCRC 符号については、例えば文献 (James S. Plank: Software-Practice & Experience, 27(9), September, pp.995-1012(1997)) で開示されている。

【 0 1 3 0 】

なお、上記の実施の形態では、DNA 又は RNA を構成するヌクレオチドは 4 種類であるため、テキストデータ TX1 をバイナリーデータ BN1 に変換する際に、表 1 に示すように各ヌクレオチドを 2 ビットのデータで表している。これに対して、ヌクレオチド (又は塩基) を表すテキストデータとして、以下のような 16 種類の文字 a ~ n (8 ビットのアスキーデータ) が使用されることがある。

【 0 1 3 1 】

- a アデニン (アデニンを含むヌクレオチドと同義、以下同様)
- c シトシン
- g グアニン
- t チミン
- u ウラシル
- m アデニン、又はシトシン

- r グアニン、又はアデニン
- w アデニン、又はチミン（若しくはウラシル）
- s グアニン、又はシトシン
- y チミン（若しくはウラシル）、又はシトシン
- k グアニン、又はチミン（若しくはウラシル）
- v アデニン、グアニン、又はシトシン
- h アデニン、シトシン、又はチミン（若しくはウラシル）
- d アデニン、グアニン、又はチミン（若しくはウラシル）
- b グアニン、シトシン、又はチミン（若しくはウラシル）
- n （アデニン、シトシン、グアニン、又はチミン（若しくはウラシル））

又は（不明若しくは他の塩基）。

【 0 1 3 2 】

この場合には、これら 1 6 種類の文字を互いに異なる 4 ビットのコードに変換し、このコードを用いてテキストデータを数値データ（バイナリーデータ）に変換してもよい。これによって、データ量を 1 / 2 にすることができる。また、将来的にヌクレオチド（塩基）の種類が増加したような場合には、これらのヌクレオチドを 5 ビット、又は 6 ビットのデータで表現するようにしてもよい。

【 0 1 3 3 】

また、上記の実施の形態では、図 7 及び図 1 0 のヌクレオチドの配列を示すテキストデータよりハッシュ関数によって要約値を算出しているが、情報量としては、それらのテキストデータは図 8 及び図 1 1 のバイナリーデータ（数値データ）と等価である。従って、これらのバイナリーデータよりハッシュ関数によってそれぞれ要約値を算出し、これらの算出結果同士を比較するようにしてもよい。バイナリーデータはテキストデータに対して 1 / 4 程度であるため、要約値を算出する時間が短縮できる利点がある。

【 0 1 3 4 】

なお、上記の実施の形態では、DNA 又は RNA 中のヌクレオチドの配列（又は塩基の配列）の情報を処理対象としているが、本発明は、遺伝子を形成するヌクレオチドの配列の情報を処理する場合にも適用できることは言うまでもない。

次に、本発明の実施の形態の他の例につき説明する。本例は、タンパク質又はペプチドを構成するアミノ酸の配列情報を処理する場合に本発明を適用したものである。

【 0 1 3 5 】

本例でも基本的に図 1 のコンピュータシステム 2 A を使用できるが、DNA のシーケンサー 4 の代わりに、タンパク質のアミノ酸の配列を決定する配列読み取り装置としてのタンパク質用のシーケンサー (protein Sequencer) が情報処理装置 1 0 に接続される点が異なっている。なお、その配列読み取り装置としては、アミノ酸の配列のデータベースも使用できる。本例でも、例えば新規の試料 G のタンパク質のアミノ酸の配列をそのシーケンサーで解読した場合に、その配列を示すテキストデータ (TX 3 とする) が情報処理装置 1 0 に供給される。本例では一文字表記を採用するものとして、n 個のアミノ酸の配列に対応するテキストデータは、n バイトの長さである。本例では、その試料 G を大腸菌として、そのテキストデータ TX 3 として、図 1 3 に示すように、上記のウェブサイト 1 から入手した大腸菌の或るタンパク質の 8 2 0 個のアミノ酸の配列を示すテキストデータを使用する。

【 0 1 3 6 】

試料 G のアミノ酸配列は配列番号 3 に示されている。図 1 3 のテキストデータは、配列番号 3 の配列から数字データを除いて、その配列を一文字表記で表したものに相当する。また、図 1 3 においては、そのテキストデータが配列方向 (アミノ酸の配列方向) に 8 行で、その配列方向に直交する非配列方向に 2 6 列の 4 文字の長さの部分テキストデータに分割されており、8 6 1 番以上のアミノ酸を示すデータの位置には 0 が表示されている。

【 0 1 3 7 】

次に、情報処理装置 1 0 は、供給されたテキストデータ TX 3 に上記の MD 5 ハッシュ関数を施して 1 2 8 ビットの要約値 AB 3 を求めると共に、そのアミノ酸の配列の数 NA 3、及び先頭と末尾との 8 個ずつのアミノ酸の配列 ST 3、SB 3 を求める。テキストデータ TX 3 に対する具体的な値は下記の通りである。

AB 3 = hex(0f66dc2b3024a9739d0e912fde12b8ba) ... (1 9)

NA3 = 820

ST3 = MRVLKFGG, SB3 = TLSWKLG V

次に、情報処理装置10は、テキストデータTX3を逆方向に並べ替えたテキストデータTXR3 (=VGLKWS・・・FKLVRM)を求め、このテキストデータTXR3のMD5ハッシュ関数による要約値ABR3、及びこのテキストデータTXR3の先頭と末尾との8個ずつのアミノ酸の配列STR3, SBR3を求める。配列STR3, SBR3は、上記の配列SB3, ST3をそれぞれ逆方向に並べ替えることによって容易に求めることができる。これらの具体的な値は以下の通りである。テキストデータTXR3の配列は、テキストデータTX3の配列に対して回文 (palindrome) の関係にあるとすることができる。

【0138】

ABR3 = hex(e895f433e1e77f84b3cadeead1a52380) ... (20)

STR3 = VGLKWSLT, SBR3 = GGFKLVRM

次に、情報処理装置10は、試料Gの名前の情報 (試料を特定する情報)、配列の数NA3、テキストデータTX3、配列ST3, SB3、要約値AB3、逆方向の配列STR3, SBR3、及び逆方向の要約値ABR3を磁気ディスク装置17のマスタファイル19に記録する。この際に、マスタファイル19を複数のファイルとして、テキストデータTX3と、それ以外のデータとを別のファイルに記録してもよい。続いて、情報処理装置10は、例えば図7と同様に図13に示すように、試料GのテキストデータTX3を配列方向 (アミノ酸の配列方向) にN行で、その配列方向に直交する非配列方向にM列の4文字の長さの部分テキストデータに分割する。なお、N, Mはそれぞれ2以上の任意の整数であり、一例としてN=16, M=13に設定できる。本例ではテキストデータTX3に例えば12文字分のダミーデータ (本例では0であるが、それ以外に例えば文字Aなども使用できる) を付加して得られる832 (=4・16・13) バイトのテキストデータ (これをTX3' と呼ぶ) を作成し、テキストデータTX3' をN=8, M=26で分割する。本例では、ヌクレオチドの配列を扱う場合と異なり、その4文字分の部分テキストデータをそのまま32ビットの変換データとして扱う。なお、この際に表2に示すように、各アミノ酸を6ビットのデータ

で表してもよいが、データ量は $3/4$ 程度になるだけであるため、本例では部分テキストデータをそのまま変換データ（数値データ）として扱う。

【 0 1 3 9 】

それについて、情報処理装置 1 0 は、その 8 行で 2 6 列の変換データに対して、図 8 の例と同様に、各列の変換データの配列方向に対する法 2^{32} ($\text{mod } 2^{32}$) のもとでの和、即ち配列方向のシンδροームを計算する。更に、各行の変換データの中で奇数番目の変換データの非配列方向に対する法 2^{32} のもとでの和、及び偶数番目の変換データの非配列方向に対する法 2^{32} のもとでの和、即ち非配列方向の 2 組のシンδροームを計算する。この例においては、シンδροームそれぞれ 3 2 ビット（4 バイト）であるため、全部のシンδροームのデータ量は、1 6 8 ($= 4 \cdot 4 2$) バイトとなる。従って、全部のシンδροームのデータ量は、全体の元のテキストデータ TX 3（8 2 0 バイト）に対してほぼ $1/4 \sim 1/5$ に減少している。

【 0 1 4 0 】

次に、情報処理装置 1 0 は、試料 G の名前の情報、配列の数 NA 3、テキストデータ TX 3、要約値 AB 3、ABR 3、及びシンδροームを磁気ディスク装置 1 7 のワーキングファイル 2 0 に記録する。この際に、ワーキングファイル 2 0 を複数のファイルとしてもよい。その後、情報処理装置 1 0 は、試料 G の名前の情報、配列の数 NA 3、配列 ST 3、SB 3、要約値 AB 3、逆方向の配列 STR 3、SBR 3、及び逆方向の要約値 ABR 3 を磁気ディスク装置 1 7 のコンテンツファイル 2 1 に記録する。更に、情報処理装置 1 0 は、コンテンツファイル 2 1 中の情報を通信ネットワーク 1 を介してコンテンツのプロバイダ 3 に送信する。これによって、コンテンツファイル 2 1 中の情報はプロバイダ 3 のサーバ内の閲覧可能なコンテンツファイル 3 1 に記録されて、第 3 者がインターネットを介して自由に閲覧できるようになる。この結果、第 3 者は、公開されている試料 G の配列の数 NA 3、及び要約値 AB 3（又は必要に応じて ABR 3）を自分の保有するアミノ酸の配列の配列数、及び要約値と比較することによって、その試料 G が自分にとって新規かどうかを判定できる。また、ユーザは、その試料 G の配列情報を複数の供給者から誤って重複して購入することを回避することができ

る。

【 0 1 4 1 】

その後、コンピュータシステム 2 A の所有者（アミノ酸情報の供給者）は、ユーザから購入要求が来るのを待つ状態となる。そして、ユーザから試料 G に対する簡易データの要求があったときには、情報処理装置 1 0 は、磁気ディスク装置 1 7 のワーキングファイル 2 0 中の試料 G のシンδροームの情報を例えば電子メールの添付ファイルとしてそのユーザに送信する。シンδροームの情報を購入したユーザは、試料 G と同じ種類の自分で解読した試料のアミノ酸の配列のシンδροームと、その購入したシンδροームとを比較することによって、相違する部分の検出及び復元を或る程度行うことができる。

【 0 1 4 2 】

一方、ユーザから完全データの要求があったときには、情報処理装置 1 0 は、ワーキングファイル 2 0 中のテキストデータ T X 3 を Z I P ファイル等の形式で圧縮し、この圧縮されたデータを例えば電子メールの添付ファイルとしてそのユーザに送信する。この際に必要に応じて、ハッシュ関数による要約値 A B 3 を同時に送信してもよい。本例によれば、簡易データ（シンδροーム）はデータ量が少ないために短時間で送信することができる。

【 0 1 4 3 】

更に、そのアミノ酸の配列情報の供給者は、ワーキングファイル 2 0 に記録した情報、即ち試料 G の名前の情報、配列の数 N A 3、テキストデータ T X 3、要約値 A B 3、A B R 3、及びシンδροームを C D - R / R W ドライブ 1 5 を介して C D - R 1 6 に記録してもよい。この C D - R 1 6 から、更に多数の C D - R O M を作製してもよく、これらの記録媒体が郵送等によってユーザに販売される。

【 0 1 4 4 】

次に、本例において、アミノ酸の配列中から所望の連続する部分的な配列を選択する方法の一例につき説明する。そのため、図 1 3 の試料 G の配列が、図 1 の表示装置 1 2 の表示画面中表示されているものとして、その表示画面の右端のエッジ部を図 1 3 のエッジ部 5 1 とする。

図 1 3 において、試料 G のアミノ酸の配列はエッジ部 5 1 の左側の表示領域に表示されており、その表示領域には図 1 のマウス 2 0 4 によって制御されるカーソル 5 2 も表示されている。この場合、図 1 3 の第 1 6 列の第 2 行～第 7 行の矩形枠で囲まれた領域 5 4 内の配列を選択するものとする、本例では先ず領域 5 4 の右端部の文字” A ” の上にカーソル 5 2 を移動して、図 1 のマウス 2 0 4 の左スイッチ 2 0 4 a を操作する。その後、カーソル 5 2 がエッジ部 5 1 から更に右方向の位置 5 3 まで仮想的に移動するように、マウス 2 0 4 を右方向に移動する。

【 0 1 4 5 】

本例では、そのようにカーソル 5 2 が一方のエッジ部に達した後にも、更にカーソル 5 2 が表示領域の外側に移動するようにマウス 2 0 4 を移動すると、そのカーソル 5 2 は、そのエッジ部に対向する他方のエッジ部からその表示領域内に現れるというスクリーン・ラッピング動作が行われる。この結果、カーソル 5 2 は、図 1 3 の表示領域の不図示の左側のエッジ部の右側に移動して、領域 5 4 の左端部の文字” K ” 上に移動して、領域 5 4 の配列が選択される。この状態で一例としてマウス 2 0 4 の右スイッチ 2 0 4 b を操作することによって、領域 5 4 の配列のコピー等を行うことができる。

【 0 1 4 6 】

次に、図 1 4 は、図 1 3 の配列の第 1 5 列～第 1 7 列の配列を示し、この図 1 4 において、第 1 6 列の第 8 行の領域 5 6 A、及びこれに続く第 1 7 列の第 1 行の領域 5 6 B の配列を選択するものとする。このとき、先ず領域 5 6 A の左端部の文字” L ” の上にカーソル 5 2 を移動して、図 1 のマウス 2 0 4 の左スイッチ 2 0 4 a を操作する。その後、カーソル 5 2 がエッジ部 5 1 から更に右下方向の位置 5 5 まで仮想的に移動するように、マウス 2 0 4 を右下方向に移動する。この結果、スクリーン・ラッピング動作によって、本例のカーソル 5 2 は、図 1 4 の表示領域の不図示の左側のエッジ部の右側に移動して、領域 5 6 B の右端部の文字” L ” 上に移動して、領域 5 6 A、5 6 B の配列が選択される。この状態で一例としてマウス 2 0 4 の右スイッチ 2 0 4 b を操作することによって、領域 5 6 A、5 6 B の配列のコピー等を行うことができる。

【 0 1 4 7 】

このように本例によれば、カーソルのスクリーン・ラッピング動作によって、マウス 2 0 4 の移動量を少なくしてアミノ酸の配列中の一連の広い領域、及び左右に離れた端部の連続する領域の配列を容易に選択することができる。同様に、ヌクレオチドの配列中から所定の部分的な領域を選択する場合にも、カーソルのスクリーン・ラッピング動作を行うことによって、選択動作を容易にかつ高速に行うことができる。

【 0 1 4 8 】

次に、カーソルのスクリーン・ラッピング動作の別の例につき図 1 7 を参照して説明する。この例では、ユーザが図 1 のマウス 2 0 4 を用いて所望のアプリケーション・プログラムを起動する際の動作につき説明する。ここでは、図 1 の表示装置 1 2 の表示領域を図 1 7 の表示領域 2 0 1 a として、表示領域 2 0 1 a の長辺方向を x 方向、短辺方向を y 方向とする。また、カーソルの座標を指定できる範囲を有効座標領域 2 0 1 b とする。この場合、カーソルの座標を表示領域 2 0 1 a の外側で、かつ有効座標領域 2 0 1 b の内側に設定すると、カーソルは表示領域 2 0 1 a のエッジ部に表示される。

【 0 1 4 9 】

図 1 7 (a) は、表示領域 2 0 1 a に表示されるプログラムリストの一例を示し、この図 1 7 (a) の表示領域 2 0 1 a には、メニューリスト 2 2 1 から選択されたプログラムの第 1 のグループリスト 2 2 2 (第 1 列)、及び第 2 のグループリスト 2 2 3 (第 2 列) が x 方向に 2 列に分けて表示されている。この表示は、メニューリスト 2 2 1 上で「プログラム」の表示 (反転している) 上をカーソル 2 2 0 が通過することによって生成される。本例では、第 2 のグループリスト 2 2 3 中のグループ G 1 6 中の或るアプリケーション・プログラムを実行したいものとして、カーソル 2 2 0 を「グループ G 1 6」の表示 (反転している) 上に移動させる。本例の表示領域 2 0 1 a では、第 2 のグループリスト 2 2 3 (第 2 列) の右側 (+x 方向) にはサブ情報を表示する余地が無いいため、グループ G 1 6 のアプリケーションリスト 2 2 4 は、グループリスト 2 2 2 (第 1 列) の左側 (-x 方向) に表示される。ここで、実行したいアプリケーション・プログラム

がアプリケーション A 3 であるとする、どのようにカーソル 2 2 0 をアプリケーションリスト 2 2 4 上に移動するかが問題となる。

【 0 1 5 0 】

即ち、単にカーソル 2 2 0 をグループ G 1 6 上から左側のグループリスト 2 2 2 上に移動すると、例えばグループ G 2 のアプリケーションリストが表示され、グループ G 1 6 のアプリケーションリスト 2 2 4 の表示が消えてしまう。そこで、本例では、カーソル 2 2 0 をグループ G 1 6 上から右方向（+ x 方向）に移動させる。そして、カーソル 2 2 0 の座標を P (m , n) とすると、更にカーソル 2 2 0 の座標が、表示領域 2 0 1 a を囲む有効座標領域 2 0 1 b の + x 方向の外側の座標 P (m 1 , n 1) となるように、図 1 のマウス 2 0 4 を右方向に移動する。

【 0 1 5 1 】

この結果、カーソル 2 2 0 は、グループ G 1 6 の表示の右側の位置から、図 1 7 (b) に示すように座標 P (0 , n 1) の位置の近傍、即ちアプリケーションリスト 2 2 4 上に移動する。この後、マウス 2 0 4 を僅かに下方向に移動して、カーソル 2 2 0 をアプリケーション A 3 の表示上に移動させた状態で、図 1 の左スイッチ 2 0 4 a をクリックすることによって、極めて短時間に、かつ容易にアプリケーション A 3 のプログラムを起動することができる。

【 0 1 5 2 】

次に、本例では、例えば図 1 7 (a) において、カーソル 2 2 0 の計算上の座標が表示領域 2 0 1 a の外部で、かつ有効座標領域 2 0 1 b の内部にある（カーソル 2 2 0 は表示領域 2 0 1 a のエッジ部に表示されている）とき、即ちカーソル 2 2 0 が不活性である（アイドリング状態にある）ときには、図 1 のマウス 2 0 4 のスイッチ 2 0 4 a , 2 0 4 b に別の機能を持たせるようにしてもよい。このようにスイッチ 2 0 4 a , 2 0 4 b に別の機能を持たせるときには、カーソル 2 2 0 の形状を変形させてもよい。一例として、そのようにカーソル 2 2 0 が不活性であるときに、左スイッチ 2 0 4 a を操作しながらマウス 2 0 4 をドラッグしたときには、表示領域 2 0 1 a の大きさを所定範囲で伸縮できるようにしてもよい。更に、上記の範囲のみならず、例えばカーソル 2 2 0 が表示領域 2 0 1 a

の輪郭（エッジ部）に対して内側に隣接する幅 L 1 の枠状の領域にあるときにも、スイッチ 2 0 4 a, 2 0 4 b に対して別の機能を持たせてもよい。

【 0 1 5 3 】

以上をまとめると、本例による情報選択方法は、複数の情報（2 2 1 ～ 2 2 3）が表示された表示領域（2 0 1 a）より、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択する情報選択方法において、移動量及び移動方向の少なくとも一方の情報を含む制御情報を生成し、この生成された制御情報に基づいてその表示領域内にその複数の情報に重畳させて移動自在にカーソル（2 2 0）（ポインタ）を表示し、このカーソルとその複数の情報の表示との位置関係に基づいて、その複数の情報の何れか、又はその複数の情報の何れかに関連する情報を選択できるようにしておき、そのカーソルをその表示領域の周縁部の第 1 の端部に移動させた状態で、更にそのカーソルをその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソルをその表示領域のその周縁部のその第 1 の端部とは異なる第 2 の端部を起点としてその表示領域内で移動させるものである。

【 0 1 5 4 】

即ち、本例のカーソル（2 2 0）は、ポインティングデバイスの制御情報に応じて、スクリーン・ラッピング方式で表示領域（2 0 1 a）中を周期的に移動する。

この結果、G U I (Graphical User Interface) 方式でコンピュータ等の各種装置を操作する際に、登録してあるアプリケーション・プログラムの個数が多い場合でも、高速にカーソルを所望のアプリケーション・プログラムの位置に移動させて、そのプログラムを起動することができる。

【 0 1 5 5 】

本例において、その表示領域（2 0 1 a）が所定の軸に関して実質的に軸対称の領域（矩形領域、又は楕円形の領域等）である場合、その第 2 の端部は、その表示領域内でその所定の軸に関してその第 1 の端部と実質的に軸対称の位置に設定されると共に、そのカーソル（2 2 0）のその第 2 の端部からの移動方向は、その制御情報によってその第 1 の端部から更にそのカーソルを移動させようとし

た方向であることが望ましい。これによって、カーソル（220）の周期的な動きが容易に予測できるため、ユーザが特に習熟することなく、直ぐにその周期的なカーソル（220）の動きを活用できる。

【0156】

また、そのカーソル（220）をその表示領域（201a）のその周縁部の第1の幅の制限領域に移動させた状態、及びそのカーソルをその制限領域から更に第2の幅（L）だけ外側（201b）に移動させるようにその制御情報を与えた状態では、そのカーソルに対してその情報の選択以外の別の機能を与えることが望ましい。通常は、その表示領域（201a）の周縁部にはアプリケーション・プログラムのアイコン等は表示されていない。そこで、そのカーソル（220）がその表示領域（201a）の周縁部に有る状態では、アプリケーション・プログラムの選択以外の機能、例えばその表示領域の伸縮機能等を持たせても、アプリケーション・プログラムの選択には実質的に影響が無いと共に、カーソル（220）（ポインティングデバイス）の用途が広がる利点がある。

【0157】

また、本例では図17に示すように、その表示領域（201a）内にその複数の情報、及びこれらの情報に関連する情報が複数列（222, 223）に表示されているときに、そのカーソル（220）がその複数列の一方の端部の列（223）の所定の情報の表示を通過しているときに、その表示領域内のその複数列の他方の端部の列（222）の外側にその所定の情報に関連する複数のサブ情報（224）を表示し、更にそのカーソルをその複数列の一方の端部の列（223）からその表示領域の外側に移動させるようにその制御情報を与えたときに、そのカーソル（220）をその表示領域のその複数列の他方の端部の列（222）に近い端部P（0, n1）を起点として、その複数のサブ情報（224）の表示上に移動させて、その複数のサブ情報の何れかを選択可能としている。

【0158】

即ち、図17に示すように、その表示領域（201a）内に表示すべきアプリケーション・プログラムの個数が多い場合には、例えばその右側の端部の列（223）のサブ情報（224）が左側の端部の列（222）の外側に表示される。

このときに、本例の周期的な移動を行うカーソル（２２０）を適用すると、そのサブ情報（２２４）中のアプリケーション・プログラムを選択するためには、ポインティングデバイスによってそのカーソル（２２０）をその列（２２３）から更に右方向に移動させるようにすればよい。これによって、アプリケーション・プログラムの個数が多く、プログラムリストが複数列になるような場合でも、GUI方式で容易にカーソルを所望のアプリケーション・プログラムの位置に移動できる。

【 0 1 5 9 】

ここで、DNA又はRNAのヌクレオチドの配列（塩基の配列）に対応するテキストデータ（又はこれを表１等に基づいて変換した数値データ）の要約値を算出するためのハッシュ関数について更に説明する。例えばハッシュ関数の演算対象を、人間のDNAのヌクレオチドの配列とすると、そのテキストデータ又は数値データのファイル（以下、「原ファイル」と言う）は１００Ｍバイト程度にも達する膨大なファイルである。そこで、本発明で使用するハッシュ関数（ハッシュ演算アルゴリズム）は、演算対象の原ファイルを分割した後の複数の分割ファイルを順次処理することによって、全体の要約値を算出する機能を持つことが望ましい。

【 0 1 6 0 】

また、ハッシュ関数は、一例として所定ビット数 m_1 （ m_1 は例えば３２，６４等）のデータを１ワードとして、所定ワード数 m_2 （ m_2 は例えば１６，３２，６４等）単位で、原ファイルの要約値を算出していく。この際に、データの処理単位は、 $m_1 \cdot m_2$ ビットとなる。例えば $m_1 = 32$ ， $m_2 = 16$ では、処理単位は５１２ビットとなる。そこで、その原ファイルを複数の分割ファイルに分割する際には、最初は $m_1 \cdot m_2$ ビットの整数倍（例えば１００００倍程度）を単位として分割していき、端数として残ったデータに所定データ（長さを表すデータ、区切りデータ等）を付加して $m_1 \cdot m_2$ ビットの整数倍のファイルとすることで、要約値の演算を効率的に実行することができる。

【 0 1 6 1 】

更に、暗号理論で使用されるハッシュ関数は、テキストデータ中のスペースコ

ード及び改行コード等も全て演算処理対象としているが、ヌクレオチド及びアミノ酸の配列情報については見やすくするために、例えば配列番号 1 ～ 3 で示すように、途中にスペースコード、順序を示す数字コード、及び改行コードを挿入する場合がある。そこで、ヌクレオチドの配列情報（アミノ酸の配列情報も同様）を演算処理対象とするハッシュ関数においては、テキストデータ中の所定コードとしてのスペースコード、数字コード、及び改行コードを無視する機能を付加することが望ましい。また、隣接する文字を” - ”（ハイフン）で分けることも考えられるが、この場合には、更に” - ”記号も無視する必要がある。

【 0 1 6 2 】

更に、原ファイルを複数の分割ファイルに分割する際には、複数の分割ファイルの順序等を示すデータ（以下、「コメントデータ」と言う）を各分割ファイルに付加することが望ましいことがある。このように分割ファイル、又は 1 つの原ファイルにコメントデータを付加する場合にも、コメントデータはハッシュ関数で無視する必要がある。そのため、例えばコメントデータは所定の開始記号（例えば / * ）及び終了記号（例えば * / ）の間に記録し、ハッシュ関数で処理する際に開始記号から終了記号までのデータは無視するようにすればよい。

【 0 1 6 3 】

また、上記の実施の形態では、例えば生物の DNA のヌクレオチドの配列（又はタンパク質のアミノ酸の配列）内の先頭の一部、及び末尾の一部の配列、並びにその配列のテキストデータの要約値をインターネット上で公開することがある。この場合には、その公開されている一部の配列と、その要約値とからそのテキストデータの内容が推定される可能性もある。これを回避するために、そのテキストデータをハッシュ関数で処理する際に、その公開されている配列を除いた部分についてのみ、そのハッシュ関数を施して要約値を求めるようにしてもよい。

【 0 1 6 4 】

次に、例えば核酸や遺伝子のヌクレオチドの配列が見易いように順序を示す数字、スペース、及び改行を含んでテキストデータとして記録されたファイル（ファイル F D 1 とする）の要約値(message digest)を計算するための方法の一例につき図 1 5 を参照して説明する。なお、以下の要約値の計算は、例えば図 1 の情

報処理装置 10 において実行される。

【0165】

図 15 において、先ずステップ 151 では、ファイル FD1 中のテキストデータから数字コード、スペースコード、及び改行コードを取り除いたテキストデータをファイル FD2 に記録する。その次のステップ 152 では、例えば MD5 ハッシュ関数を用いてファイル FD2 中のテキストデータの 128 ビットの要約値を算出する。この方法は処理は単純であるが、ファイル FD1 が例えば 100M バイト程度であるとする、ファイル FD2 もほぼ 100M バイト程度になるため、記憶装置の容量を大きくする必要がある。

【0166】

MD5 ハッシュ関数のアルゴリズムについては、上記のウェブサイト 2 で詳細に開示されているが、ここでそのアルゴリズムについて簡単に説明する。

先ず、ここでは 1 ワード "word" とは、32 ビットの量であり、1 バイト "byte" とは、8 ビットの量である。そして、一列のビットは、自然に一列のバイトと解釈することができ、ここではそれぞれ 8 ビットのデータの集まりを、MSB (most significant bit)、即ち上位ビットが最初に表示される 1 バイトのデータとして解釈することができる。同様に、一列のバイトは、一列の 32 ビットのワードと解釈することができ、ここではそれぞれ 4 バイトのデータの集まりを、LSB (least significant byte)、即ち下位バイトが最初に表示される 1 ワードのデータとして解釈することができる。

【0167】

また、次のように演算等を定義する。即ち、" x_i " は x に下付き文字 i を付加した表現を意味し、その下付き文字が一つの式であるときには、例えば " x_{i+1} " のようにその式を括弧で囲むものとする。同様に、上付き文字（べき乗）としては x^i を用いる。従って、" x^i " は x の i 乗を意味する。

また、記号 "+" は、ワードの加算、即ち法 2^{32} の加算を意味する。そして、" $X \lll s$ " は、 X を s ビットだけ左側に循環的にシフトして（回転して）得られる 32 ビットの値を意味する。また、 $\text{not}(X)$ は、 X のビット毎の補数 (complement) を意味し、" $X \vee Y$ " は、 X と Y とにビット毎の OR 演算を施して得られる

値を意味し、"X xor Y"はX とY とにビット毎のXOR（排他的論理和）演算を施して得られる値を意味し、"XY"はX とY とにビット毎のAND演算を施して得られる値を意味する。

【0 1 6 8】

次に、上記のファイルFD2に記録されているテキストデータ（ファイルFD1から数字コード、スペースコード、及び改行コードを取り除いたテキストデータ）を、要約値を求めるべきbビットのメッセージであるとする。その値bは、任意の非負整数であり、bは0であってもよい。その値bは8の倍数である必要はなく、更に任意に大きい値であってもよい。そのbビットのメッセージの一連のビットは次のように表すことができる。

【0 1 6 9】

$$m_0 \ m_1 \ \dots \ m_{[b-1]}$$

そのメッセージの要約値は、次の5つのステップA～Eの処理で計算することができる。

〔ステップA〕（追加ビットの付加）

そのメッセージには、そのメッセージをビット列で表現したときの長さが法512のもとで448に合同となるように追加ビットが付加（拡張）される。即ち、そのメッセージは、その長さが512の倍数のビットよりも64ビットだけ少ない長さになるように拡張される。追加ビットの付加は、たとえそのメッセージの長さが既に法512のもとで448に合同である場合でも常に実行される。

【0 1 7 0】

追加ビットの付加は、単一のビット"1"を付加した後に、ビットの長さが法512のもとで448に合同となるようにビット"0"を付加することによって実行される。全ての場合に、少なくとも1ビット、そして最大で512ビットが付加される。

〔ステップB〕（長さ情報の付加）

そのメッセージのビット数であるb（ステップAにおける追加ビットの付加が行われる前の長さ）の64ビットの表現が、ステップAで得られたメッセージに付加される。実際には起こりそうもないが、仮にbが64ビットよりも大きいと

きには、bの表現の下位の64ビット分だけが付加される。これらのビットは、2つの32ビットのワードとして、上述の内容に対応して下位のワードが最初になるように付加される。

【0 1 7 1】

このようにして得られたメッセージのビット表現の長さは、正確に512の倍数、即ち512ビットの倍数となる。言い換えると、このようにして得られたメッセージの長さは、正確に16個の(32ビットの)ワードの倍数となる。そこで、このようにして得られたメッセージの各ワードを $M[0 \dots N-1]$ とする。ここで、Nは16の倍数である。

【0 1 7 2】

[ステップC] (要約値バッファの初期化)

要約値を計算するために4ワードのバッファ(A,B,C,D)を使用する。ここで、A, B, C, D はそれぞれ32ビットのレジスタであり、これらのレジスタは下位バイトを最初に記載する16進表現で次の値に初期化される。

ワード A: 01 23 45 67

ワード B: 89 ab cd ef

ワード C: fe dc ba 98

ワード D: 76 54 32 10

[ステップD] 16ワードブロック毎のメッセージの処理

ここでは、それぞれ入力として3個の32ビットのワードを受け取って出力として1個の32ビットのワードを生成する4個の補助的な関数を次のように定義する。

【0 1 7 3】

$$F(X,Y,Z) = XY \vee \text{not}(X) Z$$

$$G(X,Y,Z) = XZ \vee Y \text{not}(Z)$$

$$H(X,Y,Z) = X \text{ xor } Y \text{ xor } Z$$

$$I(X,Y,Z) = Y \text{ xor } (X \vee \text{not}(Z))$$

各ビット位置で、関数Fは、Xが真ならばYで、そうでなければZという条件式として作用する。関数Fは、 \vee の代わりに+を使って定義することもできた。

なぜなら、 XY と $\text{not}(X)Z$ とは、同じビット位置で共に 1 となることが決してないからである。

【0 1 7 4】

関数 G , H , I は、 X, Y, Z のビットからビット毎に並行に出力を生成する点で関数 F と同様である。

また、関数 H は、その入力に対してビット毎の XOR 演算、又はパリティ演算を施す関数である。

更にこのステップ D では、正弦関数から導かれる 64 個の要素を持つテーブル $T[1 \dots 64]$ を用いる。即ち、そのテーブルの i 番目の要素を $T[i]$ として、 i の単位をラジアンとすると、次のようになる。

【0 1 7 5】

$T[i] = \{4294967296 \times \text{abs}(\sin(i))\}$ の整数部

なお、 $\text{abs}(\sin(i))$ は $\sin(i)$ の絶対値である。これらの関数及びテーブルを用いて以下の演算を行う。

各 16 ワードのブロックを処理するために、変数 i について 0 から $(N/16 - 1)$ まで以下の「 i に関するループの始まり」から「 i に関するループの終わり」までの処理を繰り返して行う。

【0 1 7 6】

「 i に関するループの始まり」

先ず変数 j について 0 から 15 まで、1 ワードのメッセージ $M[i*16+j]$ を $X[j]$ にコピーする。

続いて、バッファ A, B, C, D の値をそれぞれ次のようにバッファ AA, BB, CC, DD にコピーする。

【0 1 7 7】

$AA = A, BB = B, CC = C, DD = D$

「ラウンド 1」

ここで、 $[abcd \ k \ s \ i]$ は次の処理を行うものと定義する。

$a = b + ((a + F(b, c, d) + X[k] + T[i]) \lll s)$

そして、次の 16 回の処理を行う。

【 0 1 7 8 】

[ABCD 0 7 1]	[DABC 1 12 2]	[CDAB 2 17 3]	[BCDA 3 22 4]
[ABCD 4 7 5]	[DABC 5 12 6]	[CDAB 6 17 7]	[BCDA 7 22 8]
[ABCD 8 7 9]	[DABC 9 12 10]	[CDAB 10 17 11]	[BCDA 11 22 12]
[ABCD 12 7 13]	[DABC 13 12 14]	[CDAB 14 17 15]	[BCDA 15 22 16]

【 ラウンド 2 】

ここで、[abcd k s i] は次の処理を行うものと定義する。

【 0 1 7 9 】

$$a = b + ((a + G(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 1 6 回の処理を行う。

[ABCD 1 5 17]	[DABC 6 9 18]	[CDAB 11 14 19]	[BCDA 0 20 20]
[ABCD 5 5 21]	[DABC 10 9 22]	[CDAB 15 14 23]	[BCDA 4 20 24]
[ABCD 9 5 25]	[DABC 14 9 26]	[CDAB 3 14 27]	[BCDA 8 20 28]
[ABCD 13 5 29]	[DABC 2 9 30]	[CDAB 7 14 31]	[BCDA 12 20 32]

【 ラウンド 3 】

ここで、[abcd k s t] は次の処理を行うものと定義する。

【 0 1 8 0 】

$$a = b + ((a + H(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 1 6 回の処理を行う。

[ABCD 5 4 33]	[DABC 8 11 34]	[CDAB 11 16 35]	[BCDA 14 23 36]
[ABCD 1 4 37]	[DABC 4 11 38]	[CDAB 7 16 39]	[BCDA 10 23 40]
[ABCD 13 4 41]	[DABC 0 11 42]	[CDAB 3 16 43]	[BCDA 6 23 44]
[ABCD 9 4 45]	[DABC 12 11 46]	[CDAB 15 16 47]	[BCDA 2 23 48]

【 ラウンド 4 】

ここで、[abcd k s t] は次の処理を行うものと定義する。

【 0 1 8 1 】

$$a = b + ((a + I(b, c, d) + X[k] + T[i]) \lll s)$$

そして、次の 1 6 回の処理を行う。

[ABCD 0 6 49]	[DABC 7 10 50]	[CDAB 14 15 51]	[BCDA 5 21 52]
---------------	----------------	-----------------	----------------

[ABCD 12 6 53] [DABC 3 10 54] [CDAB 10 15 55] [BCDA 1 21 56]
 [ABCD 8 6 57] [DABC 15 10 58] [CDAB 6 15 59] [BCDA 13 21 60]
 [ABCD 4 6 61] [DABC 11 10 62] [CDAB 2 15 63] [BCDA 9 21 64]

次に、バッファA,B,C,D の値にそれぞれ次のようにバッファAA,BB,CC,DD の値
 (このブロックの処理が始まる前のバッファA,B,C,D の値) を加算する。

【 0 1 8 2 】

$A = A + AA, B = B + BB, C = C + CC, D = D + DD$

[i に関するループの終わり]

[ステップ E] 出力

出力として計算された要約値はバッファA, B, C, D の値そのものである。即
 ち、バッファA の下位バイトから始まって、バッファD の上位バイトで終わる値
 がその要約値である。なお、要約値が32ビット又は64ビットでよいような場
 合には、それぞれ例えばバッファA、又はバッファA,B の値のみを要約値として
 用いてもよい。

【 0 1 8 3 】

また、MD5ハッシュ関数は、元のデータの推定が困難となるように複雑な処
 理を行っているが、ヌクレオチドやアミノ酸の配列データの要約値を計算する場
 合には元のデータが或る程度推定されても特に不都合がないことがある。この場
 合には、メッセージに対応する一連の所定ビット数 s (s はMD5ハッシュ関数
 では512) のブロック B_i ($i = 1 \sim I$) 毎の演算を、順次次のような簡単な
 演算で行うことも考えられる。

【 0 1 8 4 】

$$M_1 = (a \cdot B_1 + b) \bmod 2^s$$

$$M_i = (M_{i-1} \cdot B_i + b) \bmod 2^s \quad (i = 2 \sim I)$$

この場合、 a, b は0以外の s ビットの数であり、 M_I が最終的な要約値とな
 る。

次に、上記のファイルFD1の要約値を計算するための別の方法につき図16
 を参照して説明する。ここではMD5ハッシュ関数を用いて要約値を計算するも
 のとする。以下の計算も一例として図1の情報処理装置10で実行される。

【0185】

先ず図16のステップ161において、ヌクレオチド自体を表すコードの個数を表す変数NX, NYの値をそれぞれ0に設定し、要約値を表す32ビットずつのバッファA, B, C, Dの値を所定の初期値（上記のステップCで設定した値）に設定し、要約値の計算対象のテキストデータを空にする。

次のステップ162において、ファイルFD1中のテキストデータの先頭から1文字分（ここでは1バイト）の文字コード（ここでは全ての種類のコードを含む意味である）を読み取り、それに続くステップ163において、読み取った文字コードが数字コード、スペースコード、又は改行コードかをチェックする。そして、読み取った文字コードが数字、スペース、改行の何れのコードでもない、即ち本例ではA～Z, a～zの何れかのコードであるときには、ステップ164に移行して、変数NXの値に1を加算すると共に、読み取った文字コードを要約値の計算対象のテキストデータに加える。続いてステップ165において、変数NX（読み取られた有効な文字コードの個数）が、要約値の計算単位である文字数 N_A に達したかどうかを調べる。本例では、 $N_A = 512 / 8 = 64$ である。

【0186】

$NX = N_A$ であるときには、ステップ166に移行して、変数NXを0に戻すと共に、変数NY（ N_A 個の文字単位のブロック数）に1を加算した後、ステップ167に移行して、計算対象のテキストデータ（ N_A 個の文字コードを含んでいる）の要約値（A, B, C, D）を計算する。これは、上記のステップDを1回実行することを意味する。その後、計算対象のテキストデータを空にしてから、動作はステップ168に移行して、ファイルFD1中に読み取り対象となる文字コードがまた有るかどうかチェックされる。また、ステップ163で読み取られた文字コードが数字、スペース、改行の何れかのコードであるときには、ステップ169で読み取った文字コードを無視した後、ステップ168に移行する。更に、ステップ165で変数NXが N_A に達していないときにも、動作はステップ168に移行する。

【0187】

そして、ステップ168において、読み取り対象となる文字コードがまだ有る

ときには、動作はステップ 1 6 2 に戻り、ファイル F D 1 中のテキストデータから次の 1 文字分の文字コードが読み取られて、以下ステップ 1 6 3 ～ 1 6 8 の動作が繰り返される。一方、ステップ 1 6 8 において、読み取り対象となる文字コードが無くなったときには、動作はステップ 1 7 0 に移行して、要約値（A，B，C，D）が計算される。この際に、変数 N X，N Y の値より読み取られた有効な文字コードの全個数が分かるため、上記のステップ A、ステップ B、ステップ D、ステップ E が実行される。得られた要約値（A，B，C，D）が最終的な要約値となる。

【 0 1 8 8 】

具体的に MD 5 ハッシュ関数を用いて図 1 5、及び図 1 6 の計算方法で、配列番号 1、2 のヌクレオチドの配列を示すテキストデータ、及び配列番号 3 のアミノ酸の配列を示すテキストデータの要約値を計算した結果は、1 6 進数表示で以下のようなになる。これらの要約値は、配列の改行方法等を変えても変化しない一定の値である。

【 0 1 8 9 】

MD 5 の要約値（配列番号 1）= hex(1c0a0b1d72e256bb10556a2fb52d28ae)

MD 5 の要約値（配列番号 2）= hex(ec8c3c9af5630f61f3d0cd2bd13b0f0d)

MD 5 の要約値（配列番号 3）= hex(164f14406ac21158e20ba72666a033ab)

この要約値の計算方法によれば、ファイル F D 1 から逐次関係の無い文字コードを取り除きながら要約値を計算しているため、記憶装置の記憶容量を殆ど増加する必要がないという利点がある。従って、ファイル F D 1 の情報量が大きくなる程、この計算方法は有利になる。また、上記のステップ 1 5 1、1 6 3 では、所定コードとしての数字コード、スペースコード、改行コードを取り除いているが、それ以外にコメント文などを取り除くようにしてもよい。

【 0 1 9 0 】

なお、本発明は上述の実施の形態に限定されず、本発明の要旨を逸脱しない範囲で種々の構成を取り得ることは勿論である。

【 0 1 9 1 】

【発明の効果】

本発明によれば、核酸や遺伝子中のヌクレオチドの配列情報、又はタンパク質やペプチド中のアミノ酸の配列情報を、それらの配列が所定の長さを超えたときに、それらの配列を示すテキストデータよりも少ないデータ量で記録することができる。従って、それらの配列情報を通信回線を介して短時間に送信することが可能となる。

【 0 1 9 2 】

また、それらの配列を示すテキストデータ、又はこれに対応する数値データの数学的な要約値を用いた場合には、膨大な長さの2つのヌクレオチドの配列同士、又は2つのアミノ酸の配列同士の同一性を少ないデータ量で高精度に確認することができる。また、同一の複数の配列情報を誤って購入することも防止できる。

【 0 1 9 3 】

また、シンδροーム情報を用いた場合には、2つのヌクレオチドの配列（又は2つのアミノ酸の配列）の間の相違する部分を少ないデータ量で容易に検出できると共に、必要に応じてその相違する部分の情報を復元することができる。従って、例えばSNP（一塩基変位多型：Single Nucleotide Polymorphism）を少ないデータ量で容易に発見することができる。

【 0 1 9 4 】

また、本発明によれば、ヌクレオチドの配列情報、又はアミノ酸の配列情報を少ないデータ量でユーザに供給できるビジネスモデルを提供することができる。この場合に、更に数学的な要約値、又はシンδροーム情報を用いることによって、ユーザが提供された配列情報と情報供給者が保持している配列情報との同一性の確認、又は相違する部分の検出や復元を容易に行うことができる。

【 0 1 9 5 】

また、本発明の要約値の計算方法によれば、例えばヌクレオチドの配列を見易くするための数値コードやスペースコードなど、又はその配列の内容を説明するためのコメント文などの所定コードを無視して、必要な情報のみの要約値を算出できるため、その所定コードの内容が変化しても、常に同一の要約値を算出できる利点がある。従って、その要約値の計算方法は、特にヌクレオチドやアミノ酸

の配列情報の要約値を算出する場合に有効である。

【図面の簡単な説明】

【図 1】 本発明の実施の形態の一例で使用されるコンピュータシステムを示す概略構成図である。

【図 2】 その実施の形態の一例で処理対象とする DNA、及びそのヌクレオチドの配列のバイナリーデータによる表現の例を示す図である。

【図 3】 その実施の形態の一例における DNA 情報の供給者の動作の一部を示すフローチャートである。

【図 4】 図 3 の動作に続く DNA 情報の供給者の動作を示すフローチャートである。

【図 5】 その実施の形態の一例における DNA 情報のユーザの動作の一部を示すフローチャートである。

【図 6】 図 5 の動作に続く DNA 情報のユーザの動作を示すフローチャートである。

【図 7】 標準試料 E (DNA) のヌクレオチド (2048 個) の配列を表すテキストデータを 4 行で 32 列の部分テキストデータ $T(i, j)$ に分割した状態を示す図である。

【図 8】 標準試料 E の変換データ $A(i, j)$ 、及びこれらから算出されるシンδροーム $C(j)$ 、 $B1(i)$ 、 $B2(i)$ を示す図である。

【図 9】 標準試料 E のシンδροーム $C(j)$ 、 $B1(i)$ 、 $B2(i)$ を示す図である。

【図 10】 試料 F (DNA) のヌクレオチド (2048 個) の配列を表すテキストデータを 4 行で 32 列の部分テキストデータ $TF(i, j)$ に分割した状態を示す図である。

【図 11】 試料 F の変換データ $AF(i, j)$ 、及びこれらから算出されるシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ を示す図である。

【図 12】 試料 F のシンδροーム $CF(j)$ 、 $B1F(i)$ 、 $B2F(i)$ 、及び復元された変換データを示す図である。

【図 13】 試料 G (タンパク質) のアミノ酸 (820 個) の配列を表すテ

キストデータを 8 行で 2 6 列の部分テキストデータに分割した状態を示す図である。

【図 1 4】 図 1 3 中の一部のデータを示す図である。

【図 1 5】 本発明の実施の形態の第 1 の要約値計算シーケンスを示すフローチャートである。

【図 1 6】 本発明の実施の形態の第 2 の要約値計算シーケンスを示すフローチャートである。

【図 1 7】 本発明の実施の形態の表示画面内でのカーソルの移動方法の一例を示す図である。

【符号の説明】

1 …通信ネットワーク、2 A, 2 B …コンピュータシステム、3 …コンテンツのプロバイダ、4 …DNA のシーケンサー、1 0 …情報処理装置、1 5 …C D - R / R W ドライブ、1 6 …C D - R、1 7 …磁気ディスク装置、1 9 …マスターファイル、2 0 …ワーキングファイル、2 1 …コンテンツファイル、3 1 …コンテンツファイル

【配列表】

SEQUENCE LISTING

<110> Omori, Satoshi

<120> Method and apparatus for recording information of nucleotide
sequence and amino acid sequence

<130> 2001A09

<140>

<141>

<150> JP 2000-117343

<151> 2000-04-19

<150> JP 2000-149122

<151> 2000-05-19

<160> 3

<170> PatentIn Ver. 2.0

<210> 1

<211> 2048

<212> DNA

<213> *Escherichia coli*

<400> 1

```
agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60
tgatagcagc ttctgaactg gttacctgcc gtgagtaaata taaaatttta ttgacttagg 120
tcactaaata ctttaaccaa tataggcata gcgcacagac agataaaaat tacagagtac 180
acaacatcca tgaaacgcat tagcaccacc attaccacca ccatcaccat taccacaggt 240
aacggtgcgg gctgacgcgt acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300
cttttttttt cgaccaaagg taacgaggta acaaccatgc gagtgttgaa gttcggcggt 360
acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420
aggcaggggc aggtggccac cgtcctctct gccccgccca aaatcaccaa ccacctggtg 480
gcgatgattg aaaaaacat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540
cgtatttttg ccgaactttt gacgggactc gccgccgcc agccgggggtt cccgctggcg 600
caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660
agtttgttgg ggcagtgtcc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720
```

atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgttate 780
 gatccggtcg aaaaactgct ggcagtgggg cattacctcg aatctaccgt cgatattgct 840
 gagtccaccc gccgtattgc ggcaagccgc attccggctg atcacatggg gctgatggca 900
 ggtttcaccg ccggtaatga aaaaggcgaa ctgggtggcg ttggacgcaa cggttccgac 960
 tactctgctg cgggtgctggc tgcctgttta cgcgccgatt gttgagagat ttggacggac 1020
 gttgacgggg tctataacctg cgacccgcgt cagggtcccc atgagaggtt gttgaagtcg 1080
 atgtcctacc aggaagcgat ggagctttcc tacttcggcg ctaaagttct tcacccccgc 1140
 accattaccc ccacgcccc gttccagatc cttgacctga ttaaaaatac cggaatcct 1200
 caagcaccag gtacgctcat tgggtgccagc cgtgatgaag acgaattacc ggtcaagggc 1260
 atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320
 gtcggcatgg cggcgcgcggt ctttgcagcg atgtcacgcg cccgtatttc cgtgggtgctg 1380
 attacgcaat catcttccga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440
 cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg cttactggag 1500
 ccgctggcag tgacggaacg gctggccatt atctcggtgg taggtgatgg tatgcgaccc 1560
 ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620
 gccattgctc agggatcttc tgaacgctca atctctgtcg tggtaaataa cgatgatgcg 1680
 accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgacaggt tatcgaagtg 1740
 tttgtgattg gcgtcgggtg cgttggcggt gcgctgctgg agcaactgaa gcgtcagcaa 1800
 agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctcgaaggct 1860
 ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920
 aaagagccgt ttaatctcgg gcgcttaatt cgcctcgtga aagaatatca tctgctgaac 1980
 ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
 cgcaagg 2048

<210> 2

<211> 2048

<212> DNA

<213> Escherichia coli

<400> 2

agcttttcat tctgactgca acgggcaata tgtctctgtg tggattaaaa aaagagtgtc 60
 tgatagcagc ttctgaactg gttacctgcc gtgagtaaata taaaatttta ttgacttagg 120
 tcactaaata ctttaaccaa tataggcata gcgcacagac agataaaaaat tacagagtac 180
 acaacatcca tgaaacgcat tagcaccacc attaccacca ccatcaccat taccacaggt 240
 aacggtgcgg gctgacgcgt acaggaaaca cagaaaaaag cccgcacctg acagtgcggg 300
 cttttttttt cgaccaaagg taacgaggta acaaccatgc gagtgttgaa gttcggcggt 360
 acatcagtgg caaatgcaga acgttttctg cgtgttgccg atattctgga aagcaatgcc 420
 aggcaggggc aggtggccac cgtcctctct gccccgccca aaatcaccaa ccacctggtg 480
 gcgatgattg aaaaaacat tagcggccag gatgctttac ccaatatcag cgatgccgaa 540
 cgtatttttg ccgaactttt gacgggactc gccgccgcc agccgggggtt cccgctggcg 600
 caattgaaaa ctttcgtcga tcaggaattt gcccaaataa aacatgtcct gcatggcatt 660
 agtttggttg ggcagtgtcc ggatagcatc aacgctgcgc tgatttgccg tggcgagaaa 720
 atgtcgatcg ccattatggc cggcgtatta gaagcgcgcg gtcacaacgt tactgttacc 780
 gatccggtcg aaaaactgct ggcatgtggg cattacctcg aatctaccgt cgatattgct 840
 gagtccaccc gccgtattgc ggcaagccgc attccggctg atcacatggt gctgatggca 900
 ggtttcaccg ccggtaatga aaaaggcgaa ctgggtggtg ttggacgcaa cggttccgac 960
 tactctgctg cgggtgctggc tgcctgttta cgcgccgatt gttgcgagat ttggacatta 1020
 tggcgcccaa cttataacct cgaccgcgt cagggtgccc atgcgagggt gttgaagtcg 1080
 atgtcctacc aggaagcgat ggagctttcc tacttcggcg cttaaagttct tcacccccgc 1140
 accattaccc ccatcgccca gttccagatc ccttgcctga ttaaaaatac cggaatcct 1200
 caagcaccag gtacgtcat tgggtgccag cgtgatgaag acgaattacc ggtcaagggc 1260
 atttccaatc tgaataacat ggcaatgttc agcgtttctg gtccggggat gaaagggatg 1320
 gtcggcatgg cggcgcgcgt ctttgcagcg atgtcacgcg cccgtatttc cgtggtgctg 1380
 attacgaat catcttccga atacagcatc agtttctgcg ttccacaaag cgactgtgtg 1440
 cgagctgaac gggcaatgca ggaagagttc tacctggaac tgaaagaagg cttactggag 1500
 ccgctggcag tgacggaacg gctggccatt atctcgggtg taggtgatgg tatgcgacc 1560
 ttgcgtggga tctcggcgaa attctttgcc gcactggccc gcgccaatat caacattgtc 1620
 gccattgctc agggatcttc tgaacgtca atctctgtcg tggtaaataa cgatgatgcg 1680

accactggcg tgcgcgttac tcatcagatg ctgttcaata ccgatcaggt tatcgaagtg 1740
 tttgtgattg gcgtcggtgg cgttggcggt gcgctgctgg agcaactgaa gcgtcagcaa 1800
 agctggctga agaataaaca tatcgactta cgtgtctgcg gtgttgccaa ctggaaggct 1860
 ctgctcacca atgtacatgg ccttaatctg gaaaactggc aggaagaact ggcgcaagcc 1920
 aaagagccgt ttaatctcgg gcgcttaatt cgcctcgtga aagaatatca tctgctgaac 1980
 ccggtcattg ttgactgcac ttccagccag gcagtggcgg atcaatatgc cgacttcctg 2040
 cgccaagg 2048

<210> 3

<211> 820

<212> PRT

<213> Escherichia coli

<400> 3

Met Arg Val Leu Lys Phe Gly Gly Thr Ser Val Ala Asn Ala Glu Arg

1 5 10 15

Phe Leu Arg Val Ala Asp Ile Leu Glu Ser Asn Ala Arg Gln Gly Gln

20 25 30

Val Ala Thr Val Leu Ser Ala Pro Ala Lys Ile Thr Asn His Leu Val

35 40 45

Ala Met Ile Glu Lys Thr Ile Ser Gly Gln Asp Ala Leu Pro Asn Ile

50 55 60

Ser Asp Ala Glu Arg Ile Phe Ala Glu Leu Leu Thr Gly Leu Ala Ala

65 70 75 80

Ala Gln Pro Gly Phe Pro Leu Ala Gln Leu Lys Thr Phe Val Asp Gln
85 90 95

Glu Phe Ala Gln Ile Lys His Val Leu His Gly Ile Ser Leu Leu Gly
100 105 110

Gln Cys Pro Asp Ser Ile Asn Ala Ala Leu Ile Cys Arg Gly Glu Lys
115 120 125

Met Ser Ile Ala Ile Met Ala Gly Val Leu Glu Ala Arg Gly His Asn
130 135 140

Val Thr Val Ile Asp Pro Val Glu Lys Leu Leu Ala Val Gly His Tyr
145 150 155 160

Leu Glu Ser Thr Val Asp Ile Ala Glu Ser Thr Arg Arg Ile Ala Ala
165 170 175

Ser Arg Ile Pro Ala Asp His Met Val Leu Met Ala Gly Phe Thr Ala
180 185 190

Gly Asn Glu Lys Gly Glu Leu Val Val Leu Gly Arg Asn Gly Ser Asp
195 200 205

Tyr Ser Ala Ala Val Leu Ala Ala Cys Leu Arg Ala Asp Cys Cys Glu
210 215 220

Ile Trp Thr Asp Val Asp Gly Val Tyr Thr Cys Asp Pro Arg Gln Val
225 230 235 240

Pro Asp Ala Arg Leu Leu Lys Ser Met Ser Tyr Gln Glu Ala Met Glu
245 250 255

Leu Ser Tyr Phe Gly Ala Lys Val Leu His Pro Arg Thr Ile Thr Pro
260 265 270

Ile Ala Gln Phe Gln Ile Pro Cys Leu Ile Lys Asn Thr Gly Asn Pro
275 280 285

Gln Ala Pro Gly Thr Leu Ile Gly Ala Ser Arg Asp Glu Asp Glu Leu
290 295 300

Pro Val Lys Gly Ile Ser Asn Leu Asn Asn Met Ala Met Phe Ser Val
305 310 315 320

Ser Gly Pro Gly Met Lys Gly Met Val Gly Met Ala Ala Arg Val Phe
325 330 335

Ala Ala Met Ser Arg Ala Arg Ile Ser Val Val Leu Ile Thr Gln Ser
340 345 350

Ser Ser Glu Tyr Ser Ile Ser Phe Cys Val Pro Gln Ser Asp Cys Val
355 360 365

Arg Ala Glu Arg Ala Met Gln Glu Glu Phe Tyr Leu Glu Leu Lys Glu
370 375 380

Gly Leu Leu Glu Pro Leu Ala Val Thr Glu Arg Leu Ala Ile Ile Ser

His Leu Leu Asn Pro Val Ile Val Asp Cys Thr Ser Ser Gln Ala Val
545 550 555 560

Ala Asp Gln Tyr Ala Asp Phe Leu Arg Glu Gly Phe His Val Val Thr
565 570 575

Pro Asn Lys Lys Ala Asn Thr Ser Ser Met Asp Tyr Tyr His Gln Leu
580 585 590

Arg Tyr Ala Ala Glu Lys Ser Arg Arg Lys Phe Leu Tyr Asp Thr Asn
595 600 605

Val Gly Ala Gly Leu Pro Val Ile Glu Asn Leu Gln Asn Leu Leu Asn
610 615 620

Ala Gly Asp Glu Leu Met Lys Phe Ser Gly Ile Leu Ser Gly Ser Leu
625 630 635 640

Ser Tyr Ile Phe Gly Lys Leu Asp Glu Gly Met Ser Phe Ser Glu Ala
645 650 655

Thr Thr Leu Ala Arg Glu Met Gly Tyr Thr Glu Pro Asp Pro Arg Asp
660 665 670

Asp Leu Ser Gly Met Asp Val Ala Arg Lys Leu Leu Ile Leu Ala Arg
675 680 685

Glu Thr Gly Arg Glu Leu Glu Leu Ala Asp Ile Glu Ile Glu Pro Val
690 695 700

Leu Pro Ala Glu Phe Asn Ala Glu Gly Asp Val Ala Ala Phe Met Ala
705 710 715 720

Asn Leu Ser Gln Leu Asp Asp Leu Phe Ala Ala Arg Val Ala Lys Ala
725 730 735

Arg Asp Glu Gly Lys Val Leu Arg Tyr Val Gly Asn Ile Asp Glu Asp
740 745 750

Gly Val Cys Arg Val Lys Ile Ala Glu Val Asp Gly Asn Asp Pro Leu
755 760 765

Phe Lys Val Lys Asn Gly Glu Asn Ala Leu Ala Phe Tyr Ser His Tyr
770 775 780

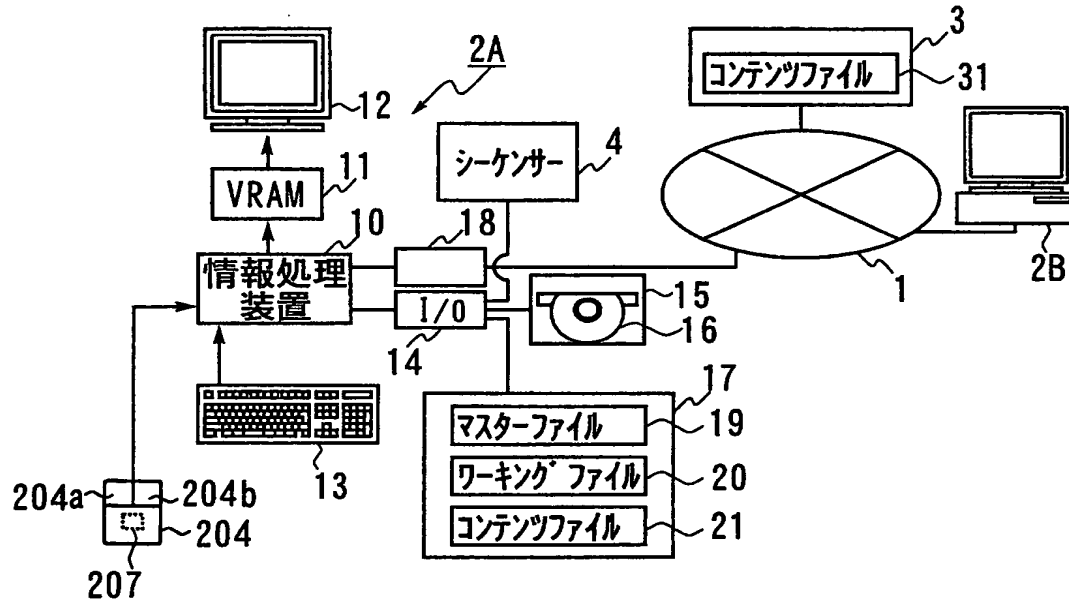
Tyr Gln Pro Leu Pro Leu Val Leu Arg Gly Tyr Gly Ala Gly Asn Asp
785 790 795 800

Val Thr Ala Ala Gly Val Phe Ala Asp Leu Leu Arg Thr Leu Ser Trp
805 810 815

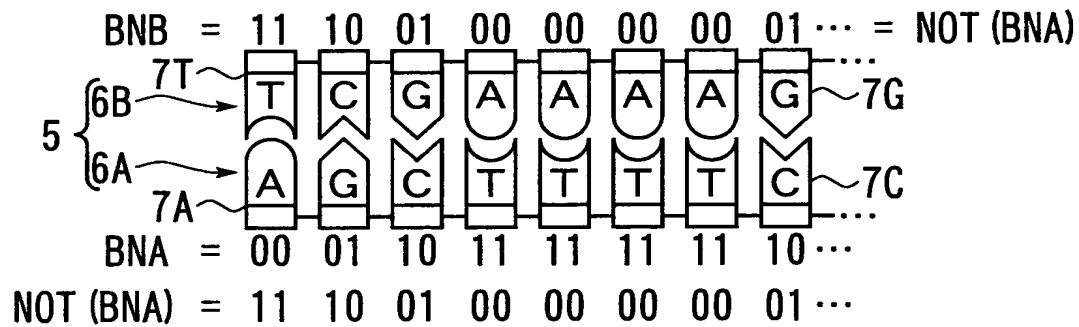
Lys Leu Gly Val
820

【書類名】 図面

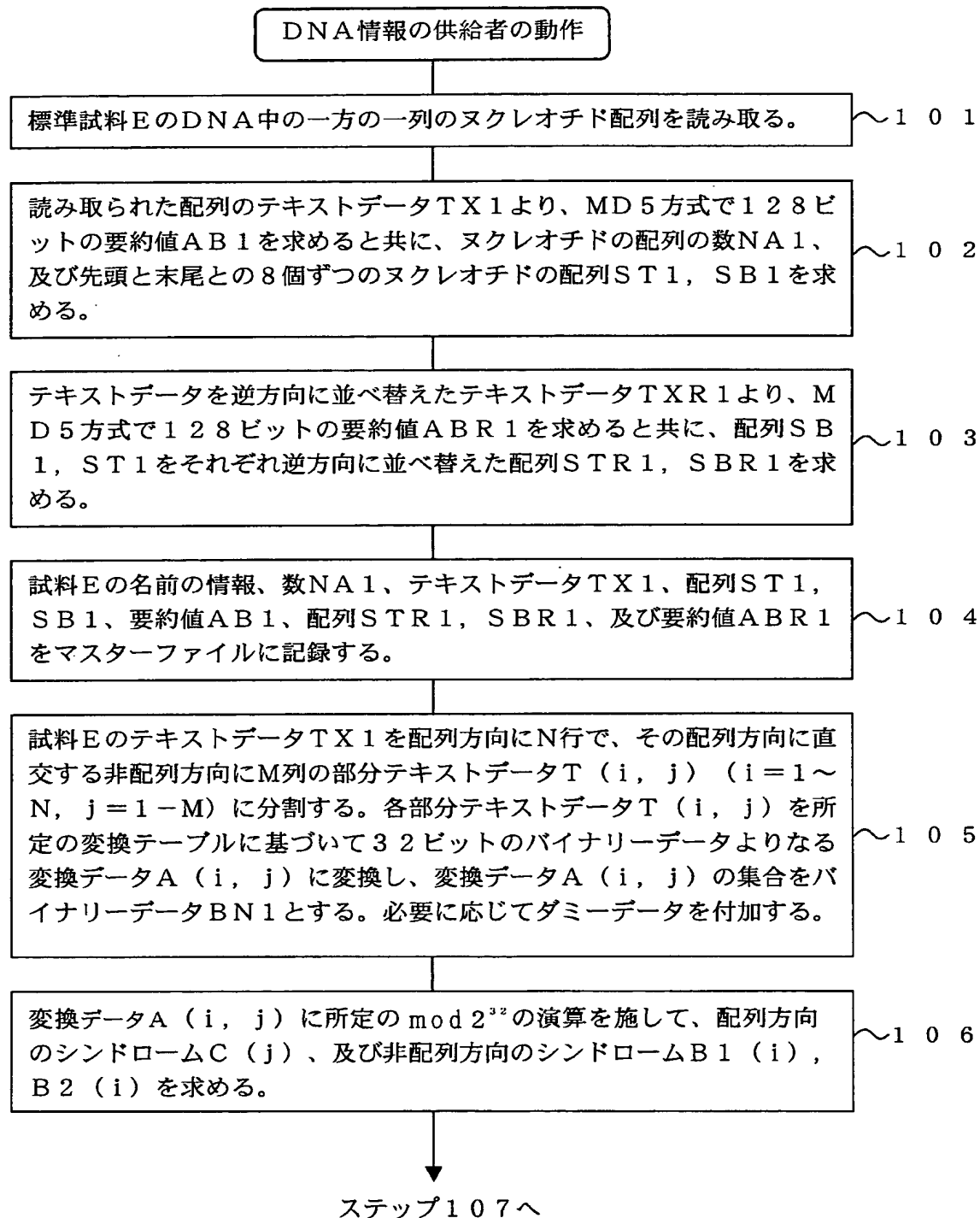
【図 1】



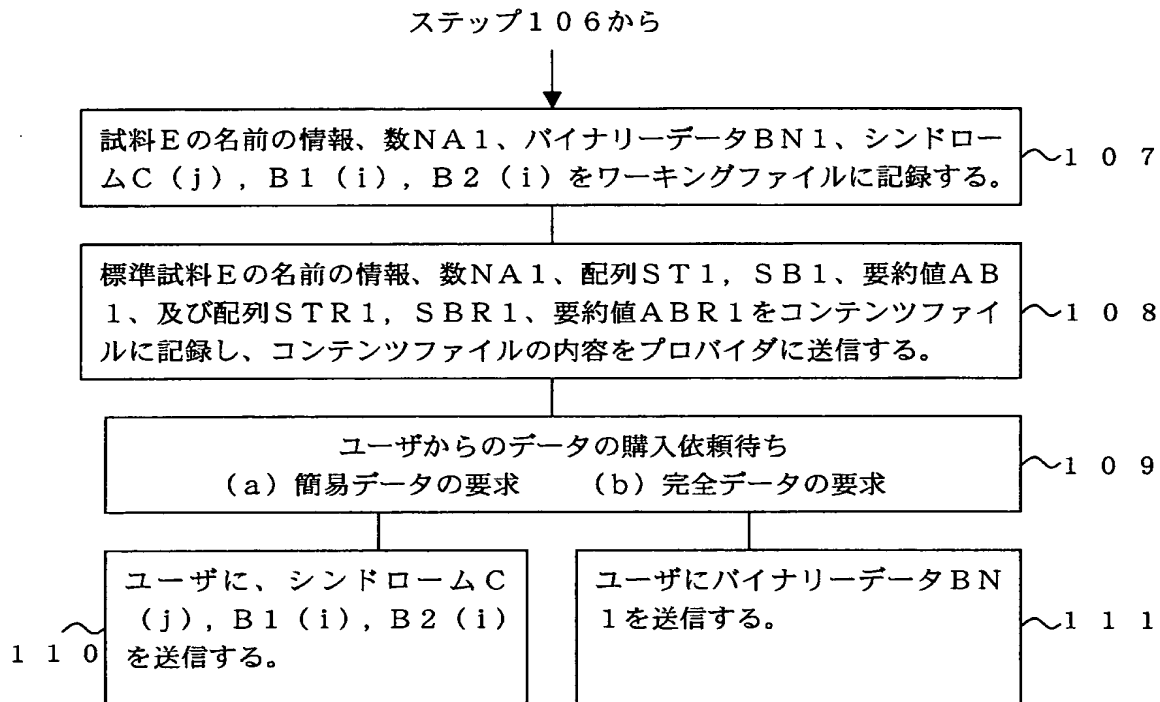
【図 2】



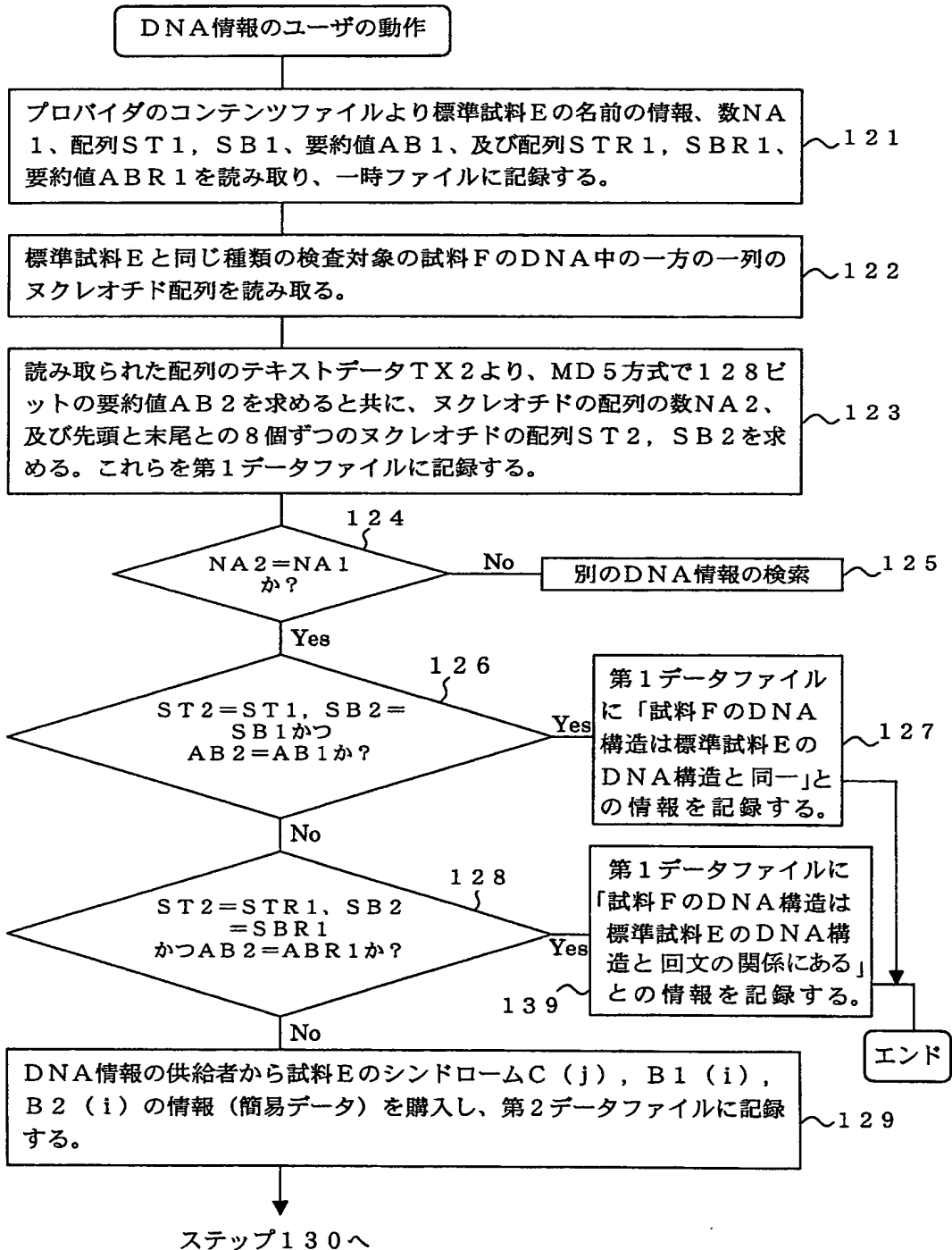
【図 3】



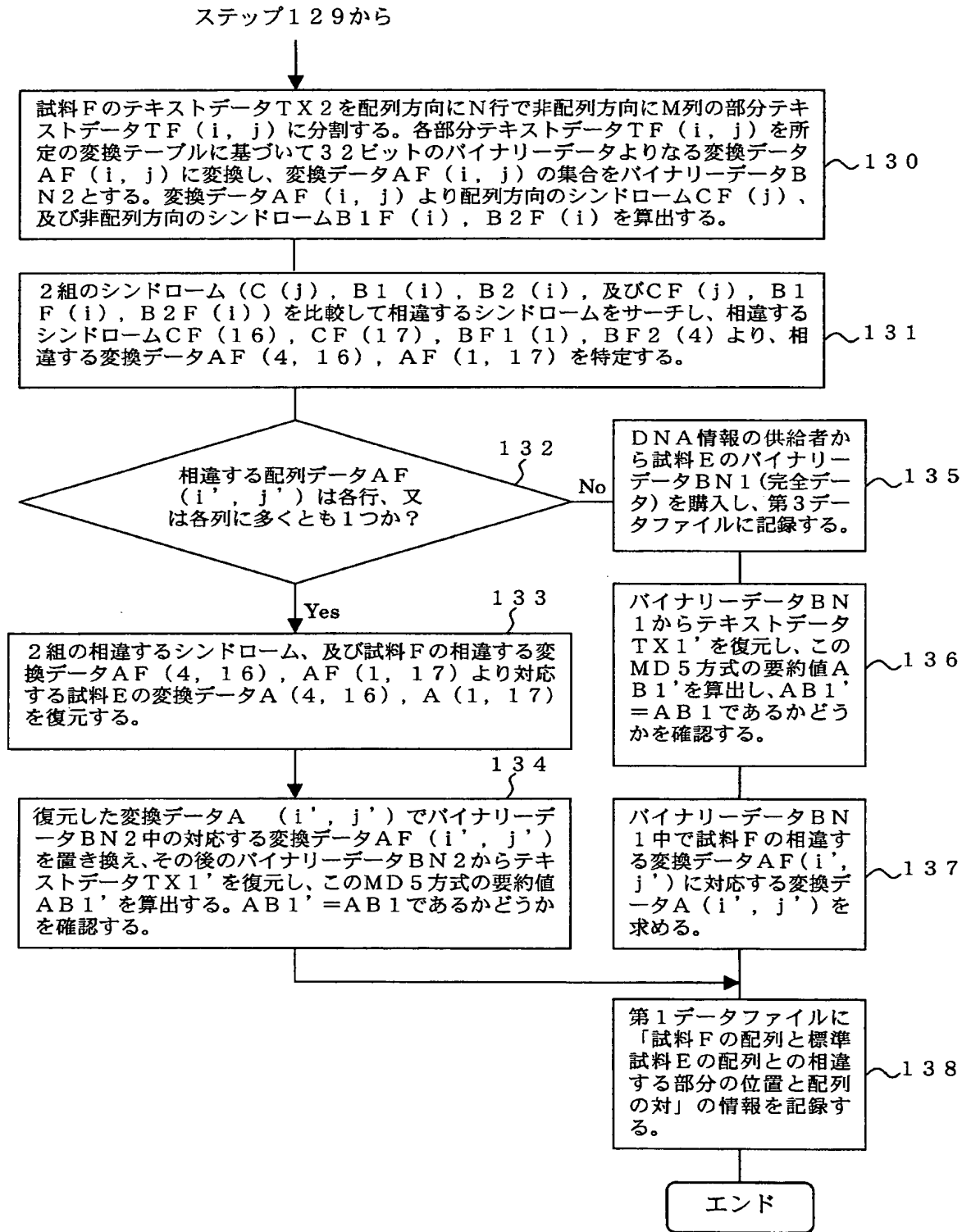
【図 4】



【图5】



【図 6】



【図 7】

標準試料 E				T(i, j)			
i	1	2	3	4			
1	AGCTTTTCATTCTGAC	TGCAACGGGCAATATG	TCTCTGTGTGGATTAA	AAAAAGAGTGTCTGAT			
2	AGCAGCTTCTGAACGTG	GTTACCTGCCGTGAGT	AAATTAAAAATTTATT	GACTTAGGTCATAAAA			
3	TACTTTAAACCAATATA	GGCATAGCGCACAGAC	AGATAAAAAATTACAGA	GTACACAACATCCATG			
4	AAACGCATTAGCACCA	CCATTACCAACCAACAT	CACCAATTACACAGGT	AACGGTCCGGGCTGAC			
5	GCGTACAGGAACACACA	GAAAAAGCCCGCACCC	TGACAGTCCGGGCTTT	TTTTTTCGACCAAAGG			
6	TAACGAGGTAAACAACC	ATCGAGTGTGGAAGT	TGCGCGGTACATCAGT	GGCAAAATGCAGAACGT			
7	TTTCTGCGTGTGCGG	ATATTCTGGAAAGCAA	TGCCAGGCAGGGGCAG	GTGGCCACCGTCTCT			
8	CTGCCCCCGCCAAAAT	CACCAACCACTGGTG	GGCATGATTGAAAAAA	CCATTAGCGGCCAGGA			
9	TGCTTTACCCCAATATC	AGCGATGCCGAACGTA	TTTTTTCGCAACTTTT	GACGGGACTCGCCGCC			
10	GCCACGCCGGGTTC	CGCTGGCGCAATTGAA	AACTTTCGTCTGATCAG	GAATTTGCCCAATAAA			
11	AACATGTCCTGCATGG	CATTAGTTTGTGGGG	CAGTGGCCGGATAGCA	TCAACGCTGGGCTGAT			
12	TTGCCGTGGCGAGAAA	ATGTCATCGCCATTAA	TGGCCGGCGTATTAGA	AGCGCGCGGTCAACAC			
13	GTTACTGTTATCGATC	CGGTCGAAAAAAGCT	GGCAGTGGGCAATTAC	CTCGAATCTACCGTCG			
14	ATATTGCTGAGTCCAC	CCGCCGTATTGCGGCA	AGCCGCATTCGGCTG	ATCACATGGTGTCTGAT			
15	GGCAGGTTTCACCGCC	GGTAATGAAAAAGCGG	AACTGGTGTCTCTGG	ACGCAACGGTTCGGAC			
16	TACTCTGCTGCGGTGC	TGGCTGCCCTGTTACG	CGCCGATTGTTGCGAG	ATTTGGACGGGACGTTG			
17	ACGGGTCTATACCTG	CGACCCCGCTCAGGTG	CCCGATGCGAGGTTGT	TGAAGTCGATGTCCTA			
18	CCAGGAAGCGATGGAG	CTTTCCTACTTTCGGCG	CTAAAGTTCTTCACCC	CCGCACCATTAACCCCC			
19	ATCGCCAGTTCAGAG	TCCCTTGCCCTGATTAA	AAATACCGGAAATCCT	CAAGCACCAAGGTACGC			
20	TCATTGGTGCCAGCCG	TGATGAAGACGAATTAA	CCGGTCAAGGGCATTT	CCAATCTGAATAACAT			
21	GGCAATGTTACAGCGTT	TCTGGTCCGGGGATGA	AAGGGATGGTCGGCAT	GGCGGCGCGCGTCTTT			
22	GCAGCGATGTCACGCG	CCCGTATTTCCTGTTG	GCTGATTACCGCAATCA	TCTTCCGAATACAGCA			
23	TCAGTTCTGCGTTCC	ACAAAGCGACTGTGTG	CGAGCTGAACGGGCAA	TGCAGGAAGAGTTCTA			
24	CCTGGAACGTGAAGAA	GGCTTACTGGAGCCGC	TGGCAGTGACGGAACG	GCTGGCCATTATCTCG			
25	GTGGTAGGTGATGGTA	TGCGCACCTTGCGTGG	GATCTCGCGGAAATTC	TTTGCCGCACTGGCCC			
26	CGCCCAATATCAACAT	TGTCGCCATTGCTCAG	GGATCTTCTGAACGCT	CAATCTCTGTCGTGGT			
27	AAATAACGATGATGCG	ACCACTGGCGTGGCGG	TTACTCATCAGATGCT	GTTCAATACCGATCAG			
28	GTTATCGAAGTGTG	TGATTGGCGTGGGTGG	CGTTGGCGGTGGGCTG	CTGGAGCAACTGAAGC			
29	GTCAGCAAAAGCTGGCT	GAAGAATAAACATATC	GACTTACGTGTCGCG	GTGTTGCCAACTCGAA			
30	GGCTCTGCTCACCAAT	GTACATGGCCCTTAATC	TGGAAAACTGGCAGGA	AGAACTGGCGCAAGCC			
31	AAGAGCCGTTTAAATC	TCGGGCGCTTAAATCG	CCCTCGTGAAAGCAATAT	CATCTGCTGAACCCGG			
32	TCATTGTTGACTGCAC	TTCCAGCCAGGCAATG	GCGGATCAATATGCCG	ACTTCTGCGCGGAAGG			

【図 8】

標準試験料 E		A(i, j)				C(j)	
i	1	2	3	4			
j	1	d82560cd	eedd4f0	0011ded3		e3135362	
1	1bfe3ed2	7cada747	03c03fcf	4bc5e2c0		e4a37e03	
2	186fb42d	58c66212	13003c84	72208e8d		a9d7cdef	
3	cbf0a0cc	a3ca28a3	8a3ca217	097656d2		39e0e7b4	
4	0263c628	4001a98a	d21d95bf	fff92805		7939a7d6	
5	67214088	36477d07	e5972387	580d8427		3631e6bf	
6	c245c20a	33ed4060	da161561	75a29ebb		827fd3e5	
7	fed9df69	8a0a2b5d	64d3d000	a3c65a14		494ebd74	
8	b6aa6803	1936909c	ffda42ff	4952e69a		3e565b03	
9	dbf2a0ce	9b5983d0	0bf9e4e1	43f6a030		55645edb	
10	6ala55fa	8f1fd155	876a5318	e09b66d3		00044f75	
11	08deb635	37939a3c	d69673c4	19997882		1e60eac2	
12	f69d6440	979002db	587558f2	b90eca79		25cbf494	
13	7cb7ce4e	a69cf658	1a63e96d	388d76d3		2d699e3a	
14	33db47a2	5c340059	0b5d76f5	26097e92		e5fad87a	
15	585fe29a	d6dadfc9	9a4f7d91	3f52527d		7c33894d	
16	cb6bd976						
17	255eccad	92a6785d	a93645f7	d07937ac		31b4c2ad	
18	a1419351	bfacbe59	b01fbc2a	a62812aa		b737027e	
19	39a87e84	eaef6b4f0	032940eb	818a1726		a9528b85	
20	e3d76869	d341243c	a5e0563f	a0ed0c23		fde5ef07	
21	5837e19f	ed7a5534	054d7963	596667bf		a46617f5	
22	61937899	a9cfe9d7	6d3c9838	efa43218		68442cc0	
23	e1fed9fa	20192ddd	91b42560	d85047ec		6c1c7523	
24	ad42d010	5bcb51a6	d61d2509	6d68f3b9		4c943a78	
25	75c5d35c	d98af675	4ee5903e	fda62d6a		9bdc8779	
26	66833823	de68f6e1	53bed09b	83bb79d7		1c667976	
27	030934d9	28b59d99	f2e384db	7e0ca4e1		9caefc2e	
28	7ce41dfd	d3d67975	9f59766d	b5182d06		a52c3ae5	
29	78601b5b	410c08ce	4bc9ded9	77da0b90		7d100e92	
30	5bb6e283	7235af0e	d402d614	10b5981a		b2a4ffbf	
31	011a7f0e	e566f0f9	ae740433	8edb42a5		23d0b6df	
32	e3df4b62	fala161d	65383369	2fad9905		72df2ded	
B1(i)	935ab0e2	f4d95e21	1891405c	f6f6e995			
B2(i)	aafa481c	e846c00e	3558b73f	43d9f669			

【図 9】

j	i	標準試料 E	2	3	4	C (j)
1						e3135362
2						e4a37e03
3						a9d7cdef
4						39e0e7b4
5						7939a7d6
6						3631e6bf
7						827fd3e5
8						494ebd74
9						3e565b03
10						55645edb
11						00044f75
12						1e60eac2
13						25cbf494
14						2d699e3a
15						e5fad87a
16						7c33894d
17						31b4c2ad
18						b737027e
19						a9528b85
20						fde5ef07
21						a46617f5
22						68442cc0
23						6c1c7523
24						4c943a78
25						9bdc8779
26						1c667976
27						9caefc2e
28						a52c3ae5
29						7d100e92
30						b2a4ffbf
31						23d0b6df
32						72df2ded
B1 (i)		935ab0e2	f4d95e21	1891405c	f6f6e995	
B2 (i)		aafa481c	e846c00e	3558b73f	43d9f669	

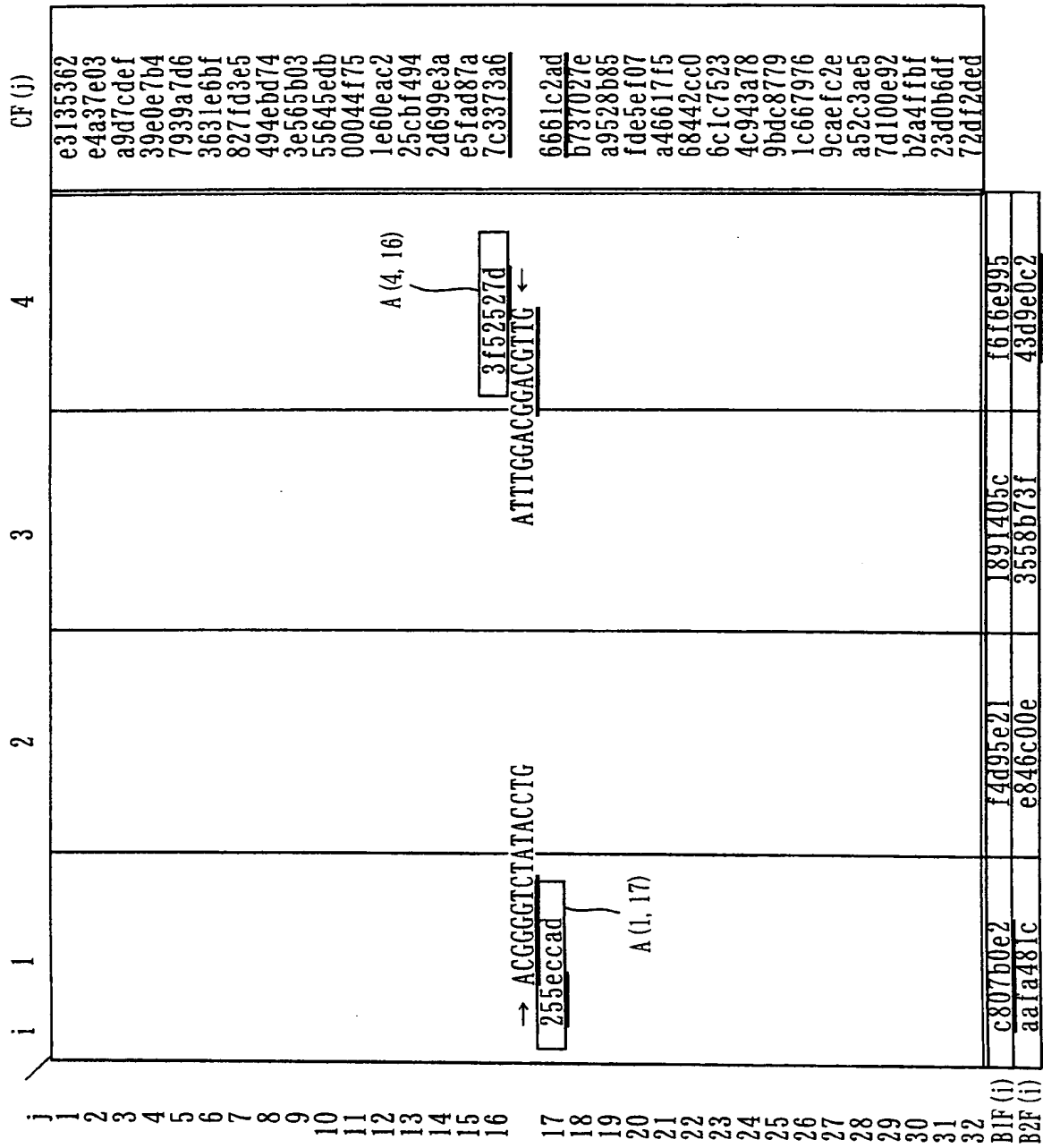
【図 10】

j	i	試料 F	TF (i, j)			
			1	2	3	4
1	1	AGCTTTTCATTTCTGAC	TGCAACGGGCAATATG	TCTCTGTGTGGATTAA	AAAAAGAGTGTCTGAT	
2	2	AGCAGCTTCTGAACTG	GTTACCTGCCGTGAGT	AAATTAATAATTTTAT	GACTTAGGTCACTAAA	
3	3	TACTTTAACCAATATA	GGCATAGCGCACAGAC	AGATAAAAATTACAGA	GTACACAAACATCCATG	
4	4	AAACGCATTAGCACCA	CCATTACCAACCAACAT	CACCATTAACACAGGT	AACGGTGCGGGCTGAC	
5	5	CGGTACAGGAACACACA	GAAAAAGCCCGCACCC	TGACAGTGGGGCTTT	TTTTTCGACCAAAGG	
6	6	TAAACGAGGTAAACAAC	ATGCGAGTGTGGAAGT	TCGGCGGTACATCAGT	GGCAAAATGCAGAACGT	
7	7	TTTCTGCGTGTGTCGG	ATATTCTGGAAAGCAA	TGCCAGGCAGGGGCAG	GTGGCCACCGTCTCT	
8	8	CTGCCCCCGCCAAAAT	CACCAACCACTGGTG	GGCATGATTGAAAAAA	CCATTAGCGGCCAGGA	
9	9	TGCTTTACCCAATATC	AGCGATGCCGAACGTA	TTTTTGCCGAACCTTT	GACGGGACTCGCCGCC	
10	10	GCCCAGCCGGGGTTCC	CGCTGGCGCAATTGAA	AACCTTCGTCGATCAG	GAAATTTGCCCAATAA	
11	11	AACATGTCCTGCATGG	CATTAGTTTGTGGGG	<u>CAGTCCCGGATAGCA</u>	TCAACGCTGCGCTGAT	
12	12	TTGCCGTGCCGAGAAA	ATGTCGATCGCCATTA	TGGCCCGCGTATTAGA	AGCGCGGGTCACAAC	
13	13	GTTACTGTTATCGATC	CGGTCGAAAAAAGTCT	GGCAGTGGGGCATTAAC	CTCGAATCTACCGTCG	
14	14	ATATTGCTGAGTCCAC	CCGCCGTATTGCGGCA	AGCCGCATTCCGGCTG	ATCACATGGTGTGAT	
15	15	GGCAGGTTTCACCGCC	GGTAATGAAAAAGCGG	AACGTGTGTGCTTGG	ACGCAACGGTTCGGAC	
16	16	TACTCTGCTCGCGTGC	TGGCTGCCCTGTTACG	CGCCGATTGTTCCGAG	ATTGGACATTATGGC	
17	17	GGCCAACTTATACCTG	CGACCCGCGTCAGGTG	CCCGATGCGAGGTTGT	TGAAGTCGATGTCTTA	
18	18	<u>CCAGCAAGCGATGGAG</u>	CTTTCCTACTTCGGCG	CTAAAGTTCTTCACCC	CCGCACCATTAACCCC	
19	19	ATCGCCAGTTCACAGA	TCCCTTGCCCTGATTAA	AAATACCGGAAATCCT	CAAGCACCAAGGTACGC	
20	20	TCATTGGTGCCAGCCG	TGATGAAGACGAATTA	CCGGTCAAGGGCATTT	CCAATCTGAATAACAT	
21	21	GGCAATGTTACGCGTT	TCTGGTCCGGGGATGA	AAGGATGGTCGGCAT	GGCGGCGCGCTCTTT	
22	22	GCAGCGATGTCACGCG	CCCGTATTTCCGTGGT	GCTGATTACGCAATCA	TCTTCGGAATACAGCA	
23	23	TCAGTTTCTCGGTTCC	ACAAAGCGACTGTGTG	CGAGCTGAACGGGCAA	TGCAGGAAGAGTTCTA	
24	24	CCTGGAACGAAAGAA	GGCTTACTGGAGCCGC	TGGCAGTACGGGAACG	GCTGGCCATTATCTCG	
25	25	GTGGTAGGTGATGGTA	TGCGCACCTTGGCTGG	GATCTCGCGGAAATTC	TTTGGCGCACTGGCCC	
26	26	GCGCCAATATCAACAT	TGTCGCCATTGCTCAG	GGATCTTCTGAACGCT	CAATCTCTGTCTGGT	
27	27	AAATAACGATGATGCG	ACCACTGGCGTGGCGG	TTACTCATCAGATGCT	GTTCAATACCGATCAG	
28	28	GTTATCGAAGTGTTTG	TGATTGGCGTCCGTGG	CGTTGGCGGTCCGCTG	CTGGAGCAACTGAAGC	
29	29	GTCAGCAAAAGCTGGCT	GAAGAATAAACATATC	GACTTACGTGCTGCG	GTGTTGCCAACTCGAA	
30	30	GGCTCTGCTCACCAAT	GTACATGGCCTTAATC	TGGAAAACTGGCAGGA	AGAACTGGCGCAAGCC	
31	31	AAAGAGCCGTTTAATC	TCGGCGCGCTTAATTCG	CCCTCGTGAAAGAAAT	CATCTGCTGAACCCGG	
32	32	TCATTGTTGACTGCAC	TTCCAGCCAGGCAGTG	GCGGATCAATATGCCG	ACTTCTGCGCGAAGG	

【図11】

i	1	試料F	2	3	AF (i, j)	4	CF (j)
1	1bfe3ed2		d82560cd	eeddd4f0	0011ded3		e3135362
2	186fb42d		7cada747	03c03fcf	4bce2c0		e4a37e03
3	cbf0a0cc		58c66212	13003c84	72208e8d		a9d7cdef
4	0263c628		a3ca28a3	8a3ca217	097656d2		39e0e7b4
5	67214088		4001a98a	d21d95bf	fff92805		7939a7d6
6	c245c20a		36477d07	e5972387	580d8427		3631e6bf
7	fed9df69		33ed4060	da161561	75a29ebb		827fd3e5
8	b6aa6803		8a0a2b5d	64d3d000	a3c65a14		494ebd74
9	dbf2a0ce		1936909c	ffda24ff	4952e69a		3e555b03
10	6ala55fa		9b5983d0	0bf9e4e1	43f6a030		55645edb
11	08deb635		8f1fd55	876a5318	e09b66d3		00044f75
12	f69d6440		37939a3c	d69673c4	19997882		1e60eac2
13	7cb7ce4e		979002db	587558f2	b90eca79		25cbf494
14	33db47a2		a69cf658	1a63e96d	388d76d3		2df99e3a
15	585fe29a		5c340059	0b5d76f5	26097e92		e5fad87a
16	cbb6d976		d6dadfc9	9a4f7d91	3f523cd6		7c3373a6
17	5a0bccad	AF (1, 17)	92a6785d	a93645f7	d07937ac		6661c2ad
18	a1419351		bface59	b01f6e2a	a628f2aa		b737027e
19	39a87e84		eaf6b4f0	032940eb	818a1726		a9528b85
20	e3d76869		d341243c	a5e0563f	a0ed0c23		fde5ef07
21	5837e19f		ed7a5534	054d7963	596667bf		a46617f5
22	61937899		a9cfe9d7	6d3c9838	efa43218		68442cc0
23	elfed9fa		20192ddd	91b42560	d85047ec		6c1c7523
24	ad42d010		5bcb51a6	d61d2509	6d68f3b9		4c943a78
25	75c5d35c		d98af675	4ee5903e	fda62d6a		9bdc8779
26	66833823		de68f6e1	53bed09b	83bb79d7		1c667976
27	030934d9		28b59d99	f2e384db	7e0ca4e1		9caefc2e
28	7ce41dfd		d3d67975	9f59766d	b5182d06		a52c3ae5
29	78601b5b		410c08ce	4bc9ded9	77da0b90		7d100e92
30	5bb6e283		7235af0e	d402d614	10b5981a		b2a4ffbf
31	011a7f0e		e566f0f9	ae740433	8edb42a5		23d0b6df
32	e3df4b62		fala161d	65383369	2fad9905		72df2ded
B1F (i)	c807b0e2		f4d95e21	1891405c	f6f6e995		
B2F (i)	aa1a481c		e846c00e	3558b73f	43d9e0c2		

【図 12】



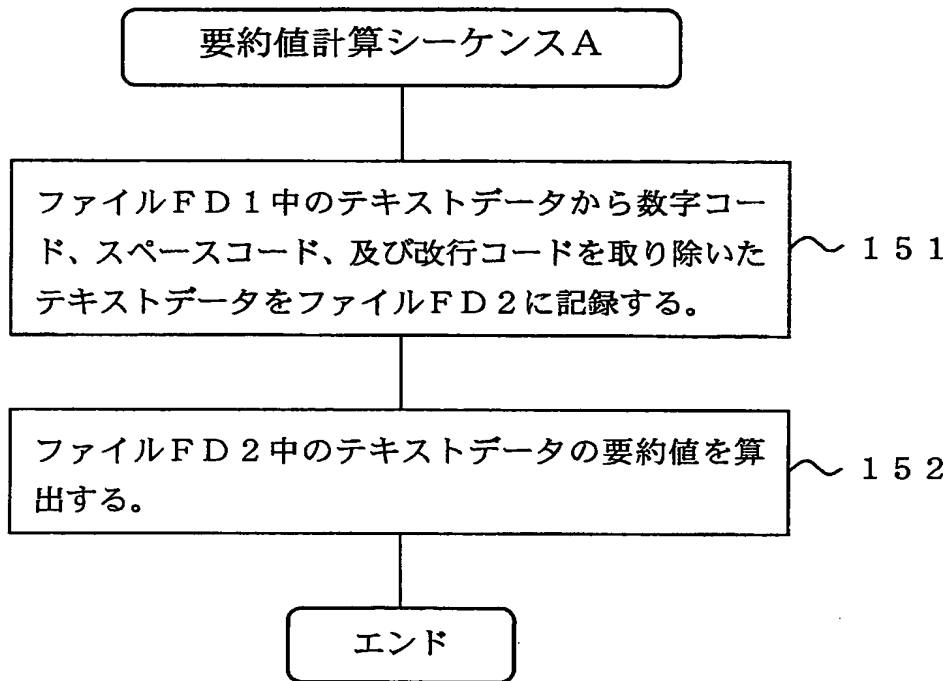
【図13】

試料G								
i	1	2	3	4	5	6	7	8
1	MRVL	KFGG	TSVA	NAER	FLRV	ADIL	ESNA	RQGQ
2	VATV	LSAP	AKIT	NHLV	AMIE	KTIS	GQDA	LPNI
3	SDAE	RIFA	ELLT	GLAA	AQPG	FPLA	QLKT	FVDQ
4	EFAQ	IKHV	LHGI	SLLG	QCPD	SINA	ALIC	RGEK
5	MSIA	IMAG	VLEA	RGHN	VTVI	DPVE	KLLA	VGHY
6	LEST	VDIA	ESTR	RIAA	SRIP	ADHM	VLMA	GFTA
7	GNEK	GELV	VLGR	NGSD	YSAA	VLAA	CLRA	DCCE
8	IWTD	VDGV	YTCD	PRQV	PDAR	LLKS	MSYQ	EAME
9	LSYF	GAKV	LHPR	TITP	IAQF	QIPC	LIKNI	TGNP
10	QAPG	TLIG	ASRD	EDEL	PVKG	ISNL	NNMA	MFSV
11	SGPG	MKGM	VGMA	ARVF	AAMS	RARI	SVVL	ITQS
12	SSEY	SISF	CVPQ	SDCV	RAER	AMQE	EFYL	ELKE
13	GLLE	PLAV	TERL	AIIS	VVGD	GMRT	LRGI	SAKF
14	FAAL	ARAN	INIV	AIAQ	GSSE	RSIS	VVVN	NDDA
15	TTGV	RVTH	QMLF	NTDQ	VIEV	FVIG	VGGV	GGAL
16	LEQL	KRQQ	SWLK	NKHI	DLRV	CGVA	NSKA	LLTN
17	VHGL	NLEN	WQEE	LAQA	KEPF	NLGR	LIRL	VKEY
18	HLLN	PVIV	DCTS	SQAV	ADQY	ADFL	REGF	HVVT
19	PNKK	ANTS	SMDY	YHQL	RYAA	EKSR	RKFL	YDTN
20	VGAG	LPVI	ENLQ	NLLN	AGDE	LMKF	SGIL	SGSL
21	SYIF	GKLD	EGMS	FSEA	TTLA	REMG	YTEP	DPRD
22	DLSG	MDVA	RKLL	ILAR	ETGR	ELEL	ADIE	IEPV
23	LPAE	FNAE	GDVA	AFMA	NLSQ	LDDL	FAAR	VAKA
24	RDEG	KVLR	YVGN	IDED	GVCR	VKIA	EVDG	NDPL
25	FKVK	NGEN	ALAF	YSHY	YQPL	PLVL	RGYG	AGND
26	VTAA	GVFA	DLLR	TLSW	KLGV	0	0	0

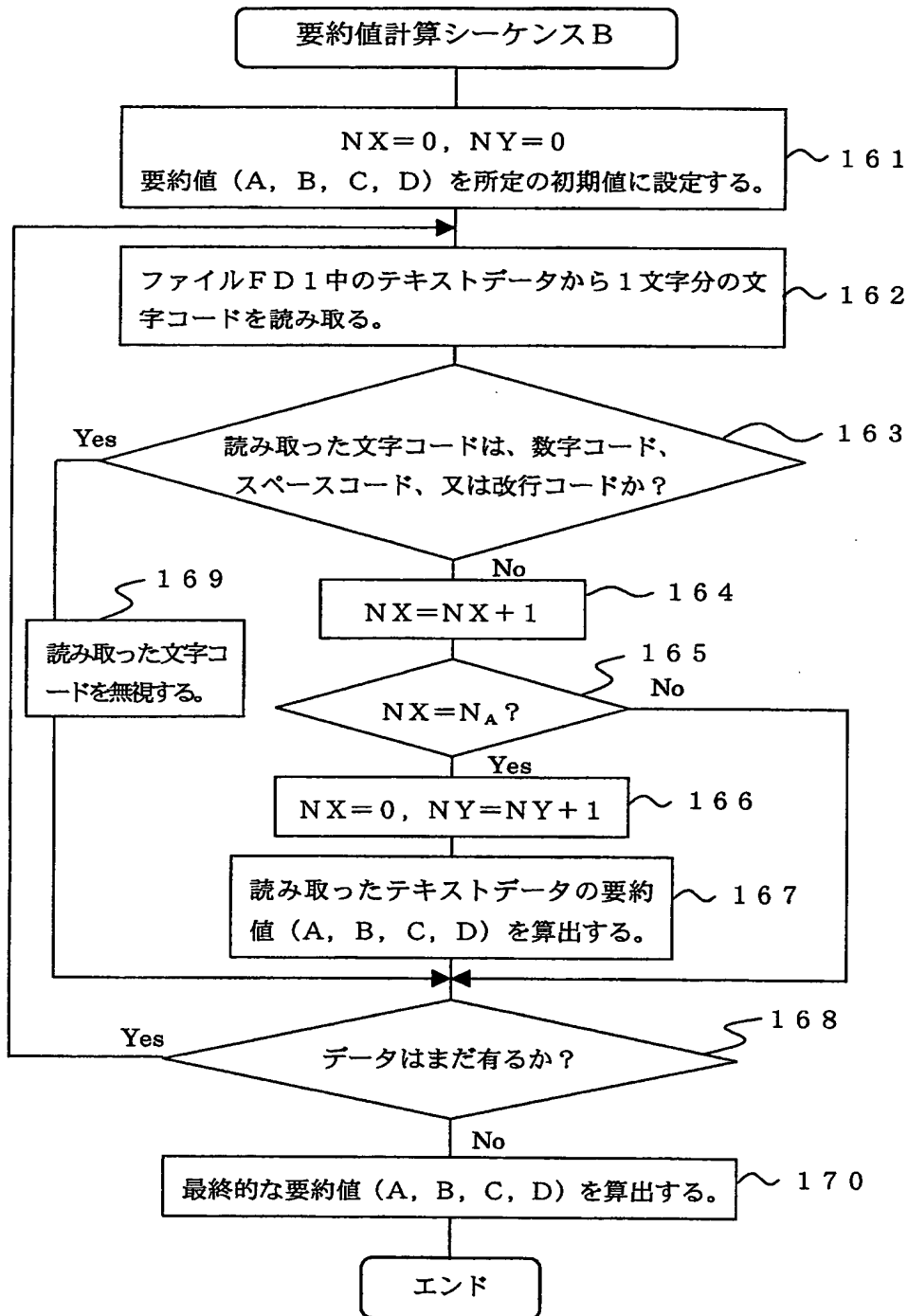
【図14】

15	TTGV	RVTH	QMLF	NTDQ	VIEV	FVIG	VGGV	GGAL	51
16	LEQL	KRQQ	SWLK	NKHI	DLRV	CGVA	NSKA	LLTN	52
17	VHGL	NLEN	WQEE	LAQA	KEPF	NLGR	LIRL	VKEY	55

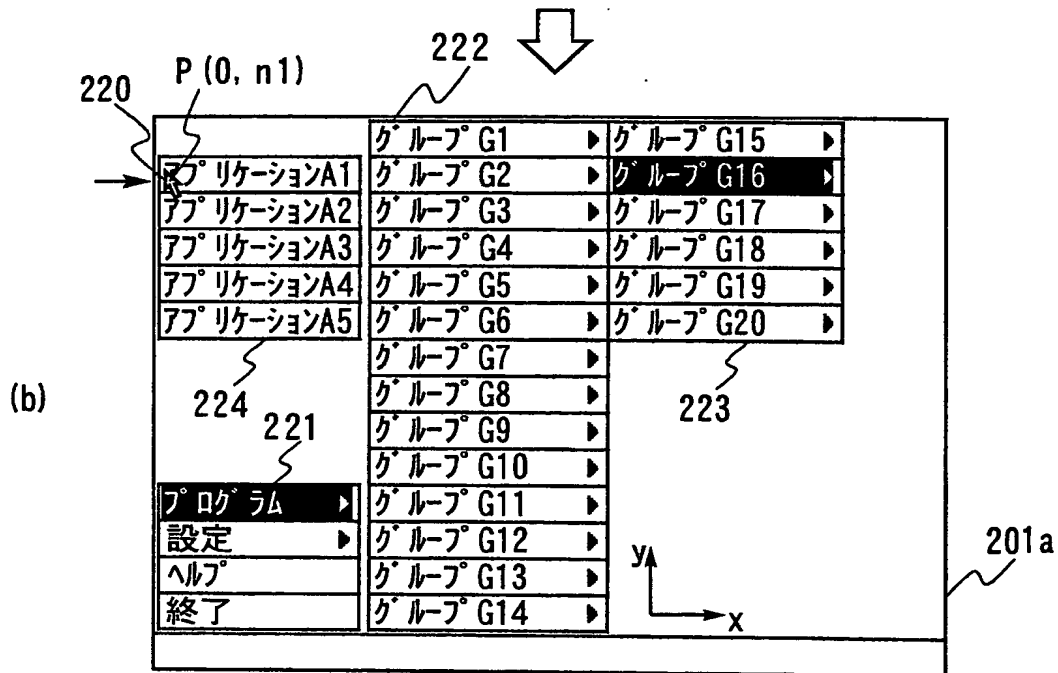
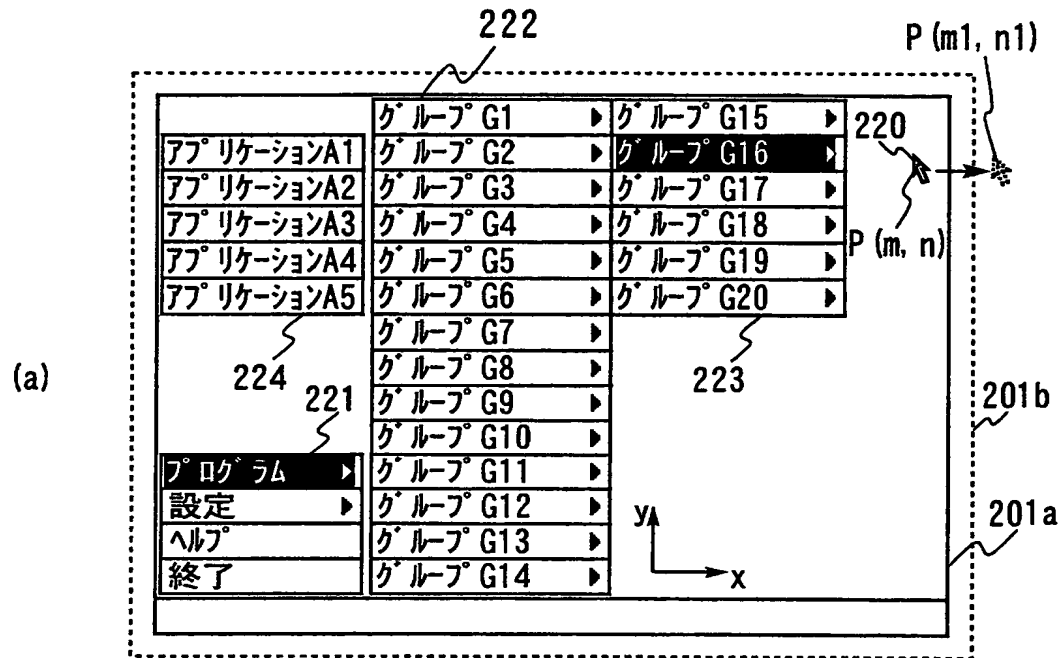
【図 1 5】



【図 16】



【図 17】



【書類名】 要約書

【要約】

【課題】 核酸中のヌクレオチドの配列情報、又はタンパク質中のアミノ酸の配列情報をできるだけ少ないデータ量で記録する。

【解決手段】 試料のDNAを構成する一方の系列のヌクレオチドの配列を示すテキストデータを所定の変換テーブルに従ってバイナリーデータに変換し、このバイナリーデータを複数行で複数列の変換データ $A(i, j)$ に分割する。各列の変換データ $A(i, j)$ を配列方向に演算してシンδροーム $C(j)$ ($j = 1, 2, \dots$)を求め、各行の変換データ $A(i, j)$ を非配列方向に演算して2組のシンδροーム $B1(i), B2(i)$ ($i = 1, 2, \dots$)を求め、シンδροーム $C(j), B1(i), B2(i)$ によってヌクレオチドの配列を近似的に表す。

【選択図】 図 8

出 願 人 履 歴 情 報

識別番号 [300000513]

1. 変更年月日 1999年12月17日
[変更理由] 新規登録
住 所 埼玉県浦和市西堀4丁目11番7号627
氏 名 大森 聡
2. 変更年月日 2001年 7月13日
[変更理由] 住所変更
住 所 埼玉県さいたま市西堀4丁目11番7号627
氏 名 大森 聡